

Request Form for Translation

U. S. Serial No.: 09/785,230

Requester's Name: Chongshan Chen

Phone No.: 305-8319

Fax No.: _____

Office Location: PK2-4B25

Art Unit/Org.: 2172

Group Director: _____

Is this for Board of Patent Appeals? No.

Date of Request: 10/6/03

Date Needed By: 10/27/03

(Please do not write ASAP-indicate a specific date)

SPE Signature Required for RUSH:

Document Identification (Select One):

(Note: Please attach a complete, legible copy of the document to be translated to this form)

1. <input checked="" type="checkbox"/> <i>RECEIVED</i>	Patent	Document No.	<u>JP 7-336659</u>
		Language	<u>JP</u>
		Country Code	<u>JP</u>
	No. of Pages	Publication Date	<u>12/22/95</u>
		(filled by STIC)	
2. <input type="checkbox"/>	TRANSLATION REQUEST LIBRARY USPTO STIC	Author	_____
	Article	Language	_____
		Country	_____
3. <input type="checkbox"/>	Other	Type of Document	_____
		Country	_____
		Language	_____

Document Delivery (Select Preference):

Delivery to Exmr. Office/Mailbox Date: _____ (*STIC Only*)

Call for Pick-up Date: _____ (*STIC Only*)

STIC USE ONLY

Copy/Search

Processor: _____

Date assigned: _____

Date filled: _____

Equivalent found: _____ (Yes/No)

Doc. No.: _____

Country: _____

Remarks: _____

Translation

Date logged in: 10-6-03

PTO estimated words: 18,600

Number of pages: 67

In-House Translation Available: _____

In-House: _____

Translator: _____

Assigned: _____

Returned: _____

Contractor: _____

Name: Flo

Priority: 1

Sent: 10/9/03

Returned: 10/27/03

Translation Branch
The world of reign prior art to you.



PTO 2004-0086

S.T.I.C. Translations Branch

Phone:	308-0881
Fax:	308-0989
Location:	Crystal Plaza 3/4 Room 2C01

To assist us in providing the most cost effective service, please answer these questions:

Will you accept an English Language Equivalent?
Yes (Yes/No)

Will you accept an English abstract?
No (Yes/No)

Would you like a consultation with a translator to review the document prior to having a complete written translation?
No (Yes/No)

Check here if Machine Translation is not acceptable:
(It is the default for Japanese Patents, '93 and onwards with avg. 5 day turnaround after receipt)

PTO 04-0086

CY=JA DATE=19951222 KIND=A
PN=07-336659

COMMUNICATIONS DEVICE
[TSUUSHIN SOUCHI]

Shuuichi Kadowaki

UNITED STATES PATENT AND TRADEMARK OFFICE
Washington, D.C. October 2003

Translated by: FLS, Inc.

PUBLICATION COUNTRY (19) : JP
DOCUMENT NUMBER (11) : 07-336659
DOCUMENT KIND (12) : A
PUBLICATION DATE (43) : 19951222
INTERNATIONAL APPLICATION NUMBER (21) : 6-125505
DATE OF FILING (22) : 19940607
ADDITION TO (61) :
INTERNATIONAL CLASSIFICATION (51) : H04N G06F 7/15 3/14 13/00 15/00
PRIORITY (30) :
INVENTORS (72) : KADOWAKI, SHUUICHI
APPLICANT (71) : CANON, INC.
DESIGNATED CONTRACTING STATES (81) :
TITLE (54) : COMMUNICATIONS DEVICE
FOREIGN TITLE [54A] : TSUUSHIN SOUCHI

(54) [TITLE OF THE INVENTION]

/1*

Communications Device

(57) [ABSTRACT]

[Purpose] To arrange so that the operation of writing the information inside a file to a shared window can be easily and quickly carried out.

[Constitution] By dragging the file icons **13 ~ 15** displayed that correspond to various files and dropping these in the desired positions inside a shared window **12**, and by arranging so that together with the expansion of the information (for example, text and images) inside the above-mentioned files inside the shared window **12**, the above-mentioned information inside the files is sent to the communications device of another party connected via a circuit so as to be shared, an editor by performing just the drag operation and drop operation with respect to the above-mentioned icons, without even activating them one by one, can take the information that corresponds to the above-mentioned icons into a shared window **12**, and the operation of taking information inside files into a shared window **12** can be easily and quickly carried out.

[CLAIMS]

/2

[Claim 1] A communications device characterized in that, in a communications device used connected to the device of the partner side via a circuit, constituted so as to mutually have a shared screen with the above-mentioned partner side device,

a means is provided so that in response to operations carried out with respect to icons displayed that correspond to files stored on the above-mentioned communications device the information in the above-mentioned files is expanded in the above-mentioned shared screen and is shared with the above-mentioned partner side device.

[Claim 2] The communications device described in Claim 1 characterized in that the above-mentioned operations are the drag

* Number in the margin indicates pagination in the foreign text.

operation and the drop operation with respect to the icons.

[Claim 3] A communications device characterized in that, in a communications device used connected to the device of the partner side via a circuit, constituted so as to mutually have a shared screen with the above-mentioned partner side device,

a click means that designates by the click operation the desired position in the above-mentioned shared window of the communications device, and

a drag means that drags a displayed file icon outside of the above-mentioned shared window from the outside to the inside of the above-mentioned shared window, and

a drop means that drops a file icon dragged by means of the above-mentioned drag means in a position inside of the designated shared window by means of the above-mentioned click means, and

an insertion means that inserts in a position designated by means of the above-mentioned click means information in a file that corresponds to the file icon dragged by means of the above-mentioned drag means after the drop operation by means of the above-mentioned drop means, and

a sending means that sends to the partner side device information inside a file that corresponds to the file icon dragged by means of the above-mentioned drag means and information of the position designated by means of the above-mentioned click means are provided.

[Claim 4] In the communications device described in Claim 3 a communications device characterized in that

a receiving means that receives information inside a file of the partner side device and information of the designated position within the shared window of the above-mentioned partner side device, and

an insertion means that inserts information inside the above-mentioned file that has been received in the designated position that has been received within the shared window of the side of one's own device that has been received by the above-mentioned receiving means are provided.

[Claim 5] The communications device described in Claims 1, 3 or 4 characterized in that the information in the file shared with the above-mentioned partner side device is text data.

[Claim 6] The communications device described in Claims 1, 3 or 4 characterized in that the information in the file shared with the above-mentioned partner side device is image data.

[Claim 7] A communications device characterized in that, in a communications device used connected to the device of the partner side via a circuit, constituted so as to mutually have a shared screen with the above-mentioned partner side device,

a first drag means that designates by means of a drag operation the desired range in the shared window of the above-mentioned communications device, and

a second drag means that drags from the outside to the inside of the above-mentioned shared window a file icon displayed outside of the above-mentioned shared window, and

a drop means that drops a file icon dragged by means of the above-mentioned second drag means in the range inside the shared window designated by means of the above-mentioned first drag means, and

a replacement means that replaces the text of the range designated by means of the above-mentioned first drag means with the text in a file that corresponds to the file icon dragged by means of the above-mentioned second drag means after the drop operation by means of the above-mentioned drop means, and

a sending means that sends to the partner side device the text inside a file that corresponds to the file icon dragged by means of the above-mentioned second drag means and the information of the range designated by means of the above-mentioned first drag means are provided.

[Claim 8] A communications device characterized in that, in the communications device described in Claim 7

a receiving means that receives text inside a file of the partner side device and information of the range designated in a shared folder of the above-mentioned partner side device, and

a replacement means that replaces, with the text in a file received by the above-mentioned receiving means, the text of the range in the shared window of the side of one's own device which the information of the designated range, that has been received by the above-mentioned receiving means, indicates is provided.

[Claim 9] A communications device characterized in that, in a communications device used connected to the device of the partner side via a circuit, constituted so as to mutually have a shared screen with the above-mentioned partner side device,

a first drag means that designates by means of a drag operation the desired range in the shared window of the above-mentioned communications device, and

a second drag means that drags from the outside to the inside of the above-mentioned shared window a file icon displayed outside of the above-mentioned shared window, and

a drop means that drops a file icon dragged by means of the above-mentioned second drag means in the range inside the shared window designated by means of the above-mentioned first drag means, and

a display means that displays, in the range designated by the above-mentioned first drag means, the image in a file that corresponds to the file icon dragged by means of the above-mentioned second drag means after the drop operation by means of the above-mentioned drop means, and

a sending means that sends to the partner side device the information of the range designated by means of the above-mentioned first drag means and the image inside a file that corresponds to the file icon dragged by means of the above-mentioned second drag means are provided.

[Claim 10] A communications device characterized in that, in the communications device described in Claim 9,

a receiving means that receives the information of a designated range inside the above-mentioned shared window of the partner side device as well as the image inside a file of the partner side device and,

a display means that displays, in the range inside a shared window of the side of one's own device that the information of a designated range received by the above-mentioned receiving means indicates, an image inside a file received by the above-mentioned receiving means are provided. /3

[Claim 11] A communications device characterized in that, in the communications device described in Claims 6 or 9,

a means which, when an image inside a file that corresponds to the above-mentioned dragged file icon is displayed in the designated range inside the shared window, enlarges or reduces that image by the same height and width, or by separate scaling factors, so that the image does not protrude from the above-mentioned designated range is provided.

[Claim 12] The communications device described in Claims 5, 6, 7 or 9 characterized in that a display control means that displays the above-mentioned file icon changed to a drag icon during the above-mentioned file icon drag operation is provided.

[Claim 13] A communications device characterized in that, in a communications device used connected to the device of the partner side

via a circuit, constituted so as to mutually have a shared screen with the above-mentioned partner side device,

a sending means that, by means of the drag operation and drop operation with respect to the file icon displayed that corresponds to the file stored in the above-mentioned communications device, sends to the partner side device the file that corresponds to the above-mentioned file icon, and

a display means that displays the above-mentioned file icon inside a shared window during the sending of the file by means of the above-mentioned sending means, and

a display erase means that erases the file icon displayed inside the above-mentioned shared window when the reception of the file sent by the above-mentioned sending means has been completed are provided.

[Claim 14] A communications device characterized in that, in a communications device used connected to the device of the partner side via a circuit, constituted so as to mutually have a shared screen with the above-mentioned partner side device,

a drag means that drags from the outside to the inside of the above-mentioned shared window a file icon displayed outside of the above-mentioned shared window, and

a drop means that drops inside a shared window a file icon dragged by means of the above-mentioned drag means, and

a display means that displays inside a shared window a file icon dragged by means of the above-mentioned drag means after the drop operation by means of the above-mentioned drop means, and

a sending means that sends to the partner side device a file that corresponds to the file icon displayed inside the above-mentioned shared window, and

a display erase means that erases the file icon displayed inside the above-mentioned shared window when the reception of the file sent by the above-mentioned sending means has been completed are provided.

[Claim 15] A communications device characterized in that, in the communications device described in Claim 13 or 14,

a display means that displays inside the above-mentioned shared window the file icon that corresponds to the above-mentioned received file when the above-mentioned file that has been sent has been received, and

a drag means that drags to the outside of the above-mentioned shared window the above-mentioned displayed file icon, and

a drop means that drops on the outside of the above-mentioned shared window the file icon dragged by means of the above-mentioned drag means, and

a save means that saves to a desired storage means a file that corresponds to the above-mentioned file icon after the drop operation by means of the above-mentioned drop means, and

a notification means that notifies the partner side device that the above-mentioned file reception has been completed are provided.

[Claim 16] A communications device characterized in that, in the communications device described in Claim 15,

the above-mentioned display means is constituted so that, inside the above-mentioned shared window, the file icon that corresponds to the above-mentioned received file is displayed in a position that is the same as that of the file icon displayed inside the shared window of the partner side device.

[Claim 17] A communications device characterized in that, in the communications device described in Claim 15,

the above-mentioned display means is constituted so that the file icon that corresponds to the above-mentioned received file is displayed in a fixed position inside the above-mentioned shared window.

[DETAILED DESCRIPTION OF THE INVENTION]

[0001] [Field of Industrial Application]

The present invention relates to a communications device constituted so that conference information comprising text and images and the like displayed in a shared window is shared with another party via a communications circuit.

[0002] [Prior Art]

Conventionally, a communications device is used connected to the device of a communications partner via a communications circuit. And, a shared window is displayed in the display part of this communications device. In this shared window, conference information comprising text and images and the like are displayed the same as with the device of the communications partner.

[0003] Furthermore, text and images and the like can also be written in a shared window, the results of this change are promptly reflected in the shared window of the other party. When the text and images inside a certain file are written to a shared window while a communications device is being used, it is arranged so that, an editor and the like is temporarily started and the file is opened, and all of the text and images in that file are selected, copied and pasted.

[0004] Here, copy and paste is the operation in which text and images selected by a drag operation and the like are copied to a temporary storage area called a clipboard and pasted inside a separate window from the clipboard. Furthermore, a drag operation is the operation in which a mouse is moved with the mouse button pressed down.

[0005] [Problem that the Invention is to Solve]

/4

However, in the above-mentioned conventional example, when the text and images inside a certain file are written to a shared window while the communications device is being used, since an editor and the like must be temporarily started, a file opened and all of the text and images in that file selected, copied and pasted, there was the weak point that the operation is complex.

[0006] The present invention, taking into consideration the above-mentioned problems, aims to simplify the operation of writing information inside a file to a shared window and to arrange so that it can be quickly carried out.

[0007] [Means for Solving the Problem]

The communications device of the present invention, in a communications device used connected to the device of the partner side via a circuit constituted so as to mutually have a shared screen with the above-mentioned partner side device, is provided with a means devised so that, in response to an operation carried out with respect to an icon displayed that corresponds to a file stored in the above-mentioned communications device, information inside the above-mentioned file is expanded in the above-mentioned shared screen and shared with the above-mentioned partner side device.

[0008] That which are regarded as other characteristics of the present invention are that, in a communications device used connected to a device of the partner side via a circuit constituted so as to have a shared screen with the above-mentioned partner side device, a sending means that sends to the partner side device, by means of performing a drag operation and a drop operation with respect to a file icon displayed that corresponds to a file stored in the above-mentioned communications device, a file that corresponds to the above-mentioned file icon, and a display means that displays the above-mentioned file icon inside a shared window while sending a file by means of the above-mentioned sending means, and a display erase means that erases the file icon displayed inside the above-mentioned shared window when reception of the file sent by means of the above-mentioned sending means has been completed are provided.

[0009] [Operation]

Since the present invention is constituted by means of the above-mentioned technical means, by performing just a simple prescribed operation with respect to a file icon displayed in the same screen, expansion in a shared screen (shared window) of information inside a file that corresponds to the above-mentioned file icon becomes possible, and, at the time of writing text and inserting an image into a shared window, the present invention is arranged so as to finish, even without separately starting an editor.

[0010] Furthermore, according to other characteristics of the present invention, when a file is sent from the sending side device to the receiving side device, the file icon that corresponds to that sent file is displayed in the sending side device, and after that, when file reception is complete at the receiving side device, the display of the above-mentioned file icon is erased and so, whether or not the sending of the file has been completed can be confirmed at the sending side.

[0011] [Embodiment]

Below, one embodiment of the communications device of the present invention is explained with reference to the drawings. First, as the first embodiment, the case in which text inside a file is expanded in a shared window is explained.

[0012] Fig. 1 is a block diagram that shows the constitution of the communications device of the present embodiment. In Fig. 1, 1 is the CPU that controls the entire present device, 2 is the ROM that

stores the program executed by the above-mentioned CPU 1, 3 is the RAM that stores the various data used by the above-mentioned CPU 1.

[0013] Next, 4 is the external memory unit composed of a magnetic disk and the like that stores the various files (text and image and the like files) used by the above-mentioned CPU 1, 5 is the final control element composed of a keyboard and mouse and the like in order to carry out the operations of the present device, and 6 is a display controller that controls the display part 7 composed of a CRT (cathode ray tube) and LCD (liquid crystal display) and the like.

[0014] Next, 8 is a circuit controller that controls the interface of a LAN (local area network) and the like, 9 is the circuit of a LAN and the like by which the present device is connected to the device of a communications partner, and 10 is the bus that connects each of the above-mentioned constituent blocks **1 ~ 6** and **8**. A communications device of the above-mentioned kind of constitution is connected via a circuit **9** to another communications device constituted in the same way.

[0015] Fig. 2 is a drawing that shows an example of the display of the screen of the display part **7** before the expansion of the text file. In Fig. 2, 11 is the entire screen of the display part **7**, 12 is the shared window (shared screen) that shares the displayed text with a partner device.

[0016] Next, 13 ~ 15 are file icons that correspond to the various text files stored in the external memory unit **4** of Fig. 1. 16 is the drag icon displayed while the drag operation (moving the mouse

while the mouse button is held down) is being carried out.

[0017] Next, 17 is the pointer that is linked to the operation of the final control element 5 of a mouse and keyboard and the like and is arbitrarily moved on the display screen 11. This pointer 17, when various operations are performed using the final control element 5, indicates the position of that operation. 18 is the cursor displayed by means of the click operation (pressing the mouse down a short time), and 19 is the hatching displayed inside the shared window 12 by means of the drag operation.

[0018] As shown in Fig. 2 (a), when one wants to expand text in a file stored in the external memory unit 4 in the shared window 12, first of all, the pointer 17 is moved to the position where one wants the text to be expanded in the shared window 12, and, by performing the click operation there, the cursor 18 is caused to be displayed.

[0019] Next, when one wants to expand inside a shared window 12 a file that corresponds to, for example, file icon 13, the pointer 17 is moved onto file icon 13 and the drag operation is started. When the drag operation starts, the drag icon 16 is displayed at the position /5 where the moving pointer 17 is during this drag.

[0020] And, when the pointer 17 is moved to the position of the cursor 18 and the drop operation (separating from the mouse button after the above-mentioned drag operation) is performed, the contents of the file that corresponds to the file icon 13 selected by means of the above-mentioned drag operation is inserted in front of the

position of the cursor **18**.

[0021] Furthermore, when the drop operation is performed, the display of the file icon **13** of the drag source and the contents of the file that corresponds to this file icon **13** remain, but the display of the drag icon **16** on the shared window **12** disappears.

[0022] Indicating the text expansion position, as shown in Fig. 2 (b), can be carried out also by having hatching **19** displayed. In this case, hatching **19** is caused to be displayed by moving the pointer **17** to the range where one wants the text to be expanded inside the shared window **12**, and performing the drag operation there.

[0023] Then, when a file icon **13** is dragged in the same way as mentioned above and this is dropped in the position of the hatching **19**, the text in the range of the hatching **19** is replaced by the contents of the file that corresponds to the file icon **13** selected by means of the above-mentioned drag operation.

[0024] Fig. 3 is a drawing that shows an example of the display of the screen of the display part **7** after the above kind of file expansion. As explained using Fig. 2, when a file icon **13** is dropped on a cursor **18**, the contents of the file that corresponds to the above-mentioned file icon **13** is inserted in front of the position of the cursor **18**. Furthermore, when a file icon **13** is dropped on the hatching **19**, the text in the range of the hatching **19** is replaced by the contents of the file that corresponds to the above-mentioned file icon **13**.

[0025] This result amounts to the text expanded in a shared window **12** as shown in Fig. 3. Furthermore, in Fig. 3, the original text before expansion is shown by "XXX...XXX" and the file inserted or replaced by expansion is shown by "AAA...AAA".

[0026] As mentioned above, the information of the position or the range (the position of the cursor **18** or the range of the hatching **19**) where the text indicated by the final control element **5** expands, as well as the text in a file that corresponds to the designated file icon **13** is sent to a partner device via a circuit **9** by means of a circuit controller **8**.

[0027] On the other hand, a shared window **12** that is the same as that of the partner side device is displayed in the display screen **11** also in the receiving side device connected via a circuit **9** to the sending side device as mentioned above. However, file icons **13 ~ 15** are not always displayed.

[0028] By means of this, when the information of the expanded position or range of the text and the text in a file are received by means of the circuit controller **8** of the receiving side device, the same text "AAA...AAA" is displayed in the same position as that of the expanded position inside the shared window **12** of the sending side device.

[0029] Fig. 4 is a drawing that shows the memory map of RAM 3 in the present embodiment. In Fig. 4, Md is the variable that stores the drag operation mode ("range designation" or "text expansion"), Tx is the variable that stores the distance from the left edge of the shared

window **12** up to the pointer **17** in character units, Ty is a variable that stores the distance from the top edge of the shared window **12** up to the pointer **17** in character units.

[0030] Next, Sx(1) is an array that stores the distance from the left side of the shared window **12** up to the cursor **18** or the front of the hatching **19** in character units, Sx(2) is an array that stores the distance from the left side of the shared window **12** up to the trailing end of the hatching **19** in character units (when the cursor **18** is displayed it is set to zero).

[0031] Next, Sy(1) is an array that stores the distance from the top edge of the shared window **12** up to the cursor **18** or front of the hatching **19**, Sy(2) is an array that stores the distance from the top edge of the shared window **12** up to the trailing end of the hatching **19** in character units (when the cursor **18** is displayed it is set to zero).

[0032] Next, Rx(1) is a variable that stores the value of field Cx(1) that stores the received array Sx(1); Rx(2) is a variable that stores the value of field Cx(2) that stores the received array Sx(2); Ry(1) is a variable that stores the value of field Cy(1) that stores the received array Sy(1); Ry(2) is a variable that stores the value of field Cy(2) that stores received array Sy(2). Furthermore, each of the above-mentioned fields Cx(1), Cx(2), Cy(1) and Cy(2) will be mentioned later.

[0033] Next, Px is a variable that stores the distance from the left edge of the display screen **11** up to the pointer **17** in pixel

units; Py is a variable that stores the distance from the top edge of the display screen **11** up to the pointer **17** in pixel units; Dx is a variable that stores the distance from the left edge of the display screen **11** up to the drag icon **16** in pixel units; Dy is a variable that stores the distance from the top edge of the display screen **11** up to the drag icon **16** in pixel units.

[0034] Fn(i) is an array that stores the file names of the files to expand; Tx(i, j) is an array that stores the text displayed in the shared window **12** (i is the column number, j is the row number); Bf(i) is the array that stores the bit maps of file icons **13 ~ 15**; Bd(i) is the array that stores the bit map of the drag icon **16**.

[0035] Furthermore, array Bf(i) is initialized by the bit maps of file icons **13 ~ 15**, and array Bd(i) is initialized by the bit map of the drag icon **16**. And, the information of these initialized bit maps is passed to the display controller **6** at the time of the display of each icon **13 ~ 16**.

[0036] Fig. 5 is a drawing that shows the format of a packet /6 of a text file sent from the circuit controller **8**. In Fig. 5, Lp is a field that stores the length of the packet; Cm is a field that stores the command class; Cx(1) is a field that stores the value of array Sx(1); Cx(2) is a field that stores the value of array Sx(2); Cy(1) is a field that stores the value of array Sy(1); Cy(2) is a field that stores the value of array Sy(2); T(i) is a field that stores the text inside file Fn(i).

[0037] Next, in Fig. 6 ~ Fig. 10, among the programs stored in ROM 2 shown in Fig. 1, flowcharts of the processes started by the CPU 1 at the time of the occurrence of each event when text is expanded are set forth.

[0038] Fig. 6 is the flowchart of the process started at the time of a mouse click. Namely, this flowchart shows the process at the time the cursor 18 of Fig. 2(a) is caused to be displayed.

[0039] In Fig. 6, first, in step P1, whether or not the class of the object presently selected read in from the display controller 6 is "text" is determined, and, if it is "text", step P2 is proceeded to. If it is not, the process ends.

[0040] In step P2, the CPU 1 instructs the display controller 6 so as to erase the display of the cursor 18 or the hatching 19 of the position which arrays Sx(1), Sx(2), Sy(1) and Sy(2) indicate. This is to erase them when the cursor 18 and the hatching 19 have already been displayed.

[0041] Next, in step P3, the position information of the pointer 17 is read to variables Tx and Ty from the display controller 6. Then, in step P4, the values of the above-mentioned variables Tx and Ty are set in arrays Sx(1) and Sy(1) respectively, and in step 5, the value zero is respectively set in arrays Sx(2) and Sy(2). Continuing, with step 6, the display controller 6 is instructed to display the cursor 18 in the position which variables Tx and Ty indicate, namely, in the position designated by the click operation.

[0042] Fig. 7 is a flowchart of the process started at the time of the commencement of the drag operation of the mouse. Namely, the process of steps P1 ~ P7 in Fig. 7 indicate the process at the time of designating the front position of the hatching **19** of Fig. 2 (b), and the process of steps P8 ~ P13, indicates the process at the time of selecting the desired file icon from among the file icons **13 ~ 15** set forth in Fig. 2 (a) and (b).

[0043] In Fig. 7, first, in step P1, whether or not the class of the object presently selected read in from the display controller 6 is "text" is determined, and, if it is "text", step P2 is proceeded to. If it is not, step 8 is proceeded to.

[0044] In step P2, the value of variable Md is set to the value which the "range designation" represents. Next, the CPU 1, in step 3, instructs the display controller **6** to erase the display of the cursor **18** or the hatching **19** of the position which arrays Sx(1), Sx(2), Sy(1) and Sy(2) indicate.

[0045] Next, in step P4, from the display controller **6** the position information of the pointer **17** is read to variables Tx and Ty in character units. Then, in step P5, the values of Tx and Ty are set in arrays Sx(1) and Sy(1), respectively, and in step P6, the values of variables Tx and Ty are set in arrays Sx(2) and Sy(2), respectively.

[0046] After that, in step P7, the display controller **6** is instructed to display hatching **19** in the range that arrays Sx(1), Sx(2), Sy(1) and Sy(2) indicate, namely, in the range designated by the drag operation, and the process ends. Furthermore, in this case,

since $Sx(1) = Sx(2)$, $Sy(1) = Sy(2)$, hatching **19** is displayed as a point at this stage.

[0047] On the other hand, in step P8, whether or not the class of the object read from the display controller **6** and currently selected is a "file icon" is determined, and if it is a "file icon", step P9 is proceeded to. If it isn't, step 14 is proceeded to

[0048] In step P9, the value of variable Md is set to the value that represents "text expansion". Next, in step P10, the position information of the pointer **17** is read to variables Px and Py in pixel units from the display controller **6**. Then, in step P11, the values of variables Px and Py are set in variables Dx and Dy , respectively.

[0049] Next, in step P12, from the display controller **6** the file name of the file to expand is read to array $Fn(i)$. Then, in step P13, the display controller **6** is instructed so as to display a drag icon **16** in the position which variables Px and Py indicate, namely, the position in which the pointer **17** exists, and the process ends.

[0050] Furthermore, when step P14 is proceeded to from step P8, the value of variable Md is set to a value that corresponds to "idle" that represents a state in which neither "range designation" nor "text expansion" are carried out, and the process ends.

[0051] Fig. 8 is a flowchart of the process of the time when the mouse moves while dragging. Namely, the process of steps P1 ~ P5 in Fig. 8 indicates the process at the time the range of the hatching **19** of Fig. 2(b) is being designated, and the process of steps P6 ~ P10 indicates the process at the time of movement of the drag icon **16** set

forth in Fig. 2(a) and (b).

[0052] In Fig. 8, first, in step P1, whether or not the value of variable Md is the value that represents "range designation" is determined, and if it is the value of "range designation", step P2 is proceeded to. If it isn't, step P6 is proceeded to.

[0053] In step P2, the display controller **6** is instructed so /7 as to erase the display of the hatching **19** of a range which the present values of arrays Sx(1), Sx(2), Sy(1) and Sy(2) indicate. Next, in step P3, from the display controller 6 the position information of the pointer **17** is read to variables Tx and Ty in character units.

[0054] Next, in step P4, the values of variables Tx and Ty are set in arrays Sx(2) and Sy(2), respectively. Then, in step P5, the display controller **6** is instructed so as to display hatching **19** in the range which the new values of arrays Sx(1), Sx(2), Sy(1) and Sy(2) set as above indicate (however, the values of arrays Sx(1), Sy(1) that indicate the point of origin of the hatching 19 are not changed), and the process ends.

[0055] On the other hand, when step P6 has been proceeded to from step P1, whether or not the value of variable Md is a value that represents "text expansion" is determined, and if it is the value of "text expansion", step P7 is proceeded to. If it is not, the process ends.

[0056] In step P7, the display controller **6** is instructed so as to erase the display of the drag icon **16** at the position which the present values of variables Dx and Dy indicate. Continuing, in step

P8, from the display controller **6** the pointer **17** position information is read to variables Px and Py in character units.

[0057] Next, in step 9, the values of variables Px and Py are set in variables Dx and Dy, respectively. Then, the display controller **6** is instructed so as to display the drag icon **16** at the position which the new values of variables Dx and Dy set as mentioned above indicate, and the process ends.

[0058] Fig. 9 is a flowchart of the process activated at the time of the drop of a mouse. In Fig. 9, first, in step P1, whether or not the value of variable Md is the value that represents "text expansion" is determined, and if it is the value of "text expansion", step P2 is proceeded to. If it isn't, the process ends.

[0059] In step P2, the display controller **6** is instructed so as to erase the display of the drag icon **16** at the position which variables Dx and Dy indicate. Continuing, in step P3, whether or not the position which variables Dx and Dy indicate is inside the shared window **12** is determined. If it is inside the shared window **12**, step P4 is proceeded to. If it isn't, the process ends.

[0060] Furthermore, here, when the shared window **12** position information is read in pixel units from the display controller **6**, and this position information that has been read satisfies the following conditions, it is regarded as "inside the shared window 12".
left edge of shared window \leq Dx \leq right edge of shared window
bottom edge of shared window \leq Dy \leq top edge of shared window

[0061] Next, in step P4, from the display controller 6 the pointer 17 position information is read to variables Tx and Ty in character units. Then, in step P5, whether or not the position which variables Tx and Ty indicate is the position which arrays Sx(1), Sx(2), Sy(1) and Sy(2) indicate or is within a range, namely, is the position of cursor 18 or is within the range of the hatching 19 is determined. Then, if it is this position or within the range, step P6 is proceeded to. If it isn't, the process ends.

[0062] Furthermore, here, shared window 12 position information is read in character units from the display controller 6 and when this read position information satisfies the following conditions it is regarded as "the position of the cursor 18 or within the range of the hatching 19".

[0063] When $Sx(2) = 0$, $Sy(2) = 0$
Tx = Sx(1) and Ty = Sy(1)
 $Sx(2) \neq 0$, $Sy(2) \neq 0$
when Ty = Sy(1),
 $Sx(1) \leq Tx \leq$ right edge of shared window
when $Sy(1) < Ty < Sy(2)$,
left edge of shared window $\leq Tx \leq$ right edge of shared window
when Ty = Sy(2),
left edge of shared window $\leq Tx \leq Sx(2)$

[0064] Next, in step P6, the display controller 6 is instructed so as to erase the display of the cursor 18 or the hatching 19 in the position or range which the arrays Sx(1), Sx(2), Sy(1) and Sy(2)

indicate. Continuing, in step P7, the text in file Fn(i) is expanded in the position or range which arrays Sx(1), Sx(2), Sy(1) and Sy(2) indicate.

[0065] Here, when Sx(2) = 0, Sy(2) = 0, the text in file Fn(i) is inserted at the position which variables Tx and Ty indicate. Furthermore, when Sx(2) ≠ 0, Sy(2) ≠ 0, the text of the range which arrays Sx(1), Sx(2), Sy(1) and Sy(2) indicate is replaced by the text in file Fn(i).

[0066] Then, in step P8, the text expansion command is stored in field Cm, and the values of arrays Sx(1), Sx(2), Sy(1), Sy(2) are stored, respectively, in fields Cx(1), Cx(2), Cy(1), Cy(2), and the text in file Fn(i) is stored in field T(i). And, the information of these various fields is sent to a partner device via the circuit controller 8, and the process ends.

[0067] Fig. 10 is a flowchart of the process activated at the time a circuit controller 8 has received a packet containing the text expansion command in field Cm. First, in step P1, the information of fields Cx(1), Cx(2), Cy(1), Cy(2) is set in arrays Rx(1), Rx(2), Ry(1), Ry(2), respectively. Next, in step P2, the information of field T(i) received from the circuit controller 8 is expanded in the position or range which arrays Rx(1), Rx(2), Ry(1), Ry(2) indicate, and the process ends.

[0068] In this way, in the present embodiment, by just performing the drag operation and drop operation with respect to the file icon 13 displayed on the same display screen 11, the text in a file that

corresponds to the above-mentioned file icon **13**, together with /8

being expanded in a shared window **12**, can be sent to a partner device.

[0069] Furthermore, in the above-mentioned first embodiment the invention is devised so that the designated range of the text is displayed by hatching **19**, but devising so that the text is displayed by underlining, bold type or italic type is also acceptable.

[0070] Next, as the second embodiment, the case of expanding in a shared window an image inside a file will be explained. Furthermore, the constitution of the communications device of the present embodiment is the same as that set forth in Fig. 1.

[0071] Fig. 11 is a drawing that shows an example of the display of the screen of the display part **7** before expansion of the file of the image, and the same reference numerals are attached to the items that are the same as those shown in Fig. 2. In Fig. 11, **21 ~ 23** are file icons, and correspond to the files of several images stored in the external memory unit **4** of Fig. 1.

[0072] Next, **24** is the cursor displayed by means of the click operation in a shape that is different from the cursor **18** (refer to Fig. 2 (a)) displayed at the time of expansion of the above-mentioned text. Furthermore, **25** is a frame displayed inside the shared window **12** by means of the drag operation.

[0073] As shown in Fig. 11 (a), when one wants to display an image in a file stored in the external memory unit **4**, first of all, the pointer **17** is moved to the position where one wants to display the image inside the shared window **12**, and the cursor **24** is caused to be

displayed there by performing the click operation.

[0074] Next, when one wants to display the image in a file that corresponds to file icon **21**, for example, inside a shared window **12**, the pointer **17** is moved onto the file icon **21** and the drag operation is initiated. When the drag operation is initiated, the drag icon **16** is displayed at the position where the moving pointer **17** is during this drag.

[0075] After that, when the pointer **17** is moved to the position of the cursor **24** and the drop operation is performed, the image in the file that corresponds to the file icon **21** selected by the above-mentioned drag operation is displayed at the position of the cursor **24**.

[0076] Furthermore, when the drop operation is performed, the display of the file icon **21** of the drag source as well as the contents of the file that corresponds to this file icon **21** remain, but the display of the drag icon **16** on the shared window **12** disappears.

[0077] Designating the display position of an image, as shown in Fig. 11 (b), can also be carried out by causing a frame **25** to be displayed. That is, the pointer **17** is moved to the range where one wants to display the image inside the shared window **12**, and the frame **25** is caused to be displayed there by performing the drag operation. The drag operation in this case is carried out in the direction of a diagonal line of the frame **25**.

[0078] Then, when the file icon **21** is dragged in the same way as mentioned above and this is dropped in the position of the frame **25**, the image that corresponds to the file icon **21** selected by means of the above-mentioned drag operation is displayed inside the range of the frame **25**.

[0079] Fig. 12 is a drawing that shows a display of the screen of the display part **7** after the above-mentioned kind of file expansion. As shown in Fig. 12, when a file icon **21** is dropped on a cursor **24** or on a frame **25**, the image **26** that corresponds to the above-mentioned file icon **21** is displayed at the position of the cursor **18** or within the range of the frame **25**.

[0080] The information of the position or the range (the position of the cursor **24** or the range of the frame **25**) that displays the image selected by the final control element **5** as mentioned above, as well as the image data in the file that corresponds to the designated file icon **13** is sent to the partner device via a circuit **9** by means of the circuit controller **8**.

[0081] On the other hand, a shared window **12** the same as that of the partner side device is displayed on the display screen **11** also in the receiving side device connected via a circuit **9** to the sending side device as mentioned above. However, file icons **21 ~ 23** are not always displayed.

[0082] Due to this, when the information of the display position or range of the image and the image data inside a file are received by

means of the circuit controller **8** of the receiving side device, the same image **26** is displayed in the same position or range as the position or range displayed inside the shared window **12** of the sending side device.

[0083] Fig. 13 is a drawing that shows the memory map in RAM 3 in the present embodiment. Furthermore, in Fig. 13, items with the same reference numerals attached as those in the memory map shown in Fig. 4 are variables and arrays that have the same content. First of all, Md' is a variable that stores the drag operation mode ("range designation" or "image expansion").

[0084] Next, $Wx(1)$ is an array that stores the distance from the left edge of the display screen **11** up to the left edge of the shared window **12** in pixel units; $Wx(2)$ is an array that stores the distance from the left edge of the display screen **11** up to the right edge of the shared window **12** in pixel units; $Wy(1)$ is an array that stores the distance from the top edge of the display screen **11** up to the top edge of the shared window **12** in pixel units; $Wy(2)$ is an array that stores the distance from the top edge of the display screen **11** up to the bottom edge of the shared window **12** in pixel units.

[0085] Next, $Sx(1)'$ is an array that stores the distance from the left edge of the shared window **12** up to the left edge of the cursor **24** or the frame **25** in pixel units; $Sx(2)'$ is an array that stores the distance from the left edge of the shared window **12** up to the right edge of the frame **25** in pixel units (when the cursor **24** is displayed,

it is set to zero).

[0086] Next, Sy(1)' is an array that stores the distance from the top edge of the shared window 12 up to the left edge of the cursor 24 or the frame 25 in pixel units; Sy(2)' is an array that stores the distance from the top edge of the shared window 12 up to the right edge of the frame 25 (when the cursor 24 is displayed, it is set to zero). /9

[0087] Next, Rx(1)' is a variable that stores the value of the storage field Fx(1) of received array Sx(1)'; Rx(2)' is a variable that stores the value of the storage field Fx(2) of the received array Sx(2)'; Ry(1)' is a variable that stores the value of the storage field Fy(1) of the received array Sy(1)'; Ry(2)' is a variable that stores the value of the storage field Fy(2) of the received array Sy(2)'. Furthermore, each of the above-mentioned storage fields Fx(1), Fx(2), Fy(1), Fy(2) will be mentioned later.

[0088] Next, Bx is a variable that stores the number of pixels with respect to the horizontal direction of the image 26 in the file displayed in the shared window 12; By is the variable that stores the number of pixels with respect to the vertical direction of the image 26 in the file displayed in the shared window 12; B(i, j) is the array that stores the bit map of the image 26 in the file displayed in the shared window 12 (i is the number of a pixel of the horizontal direction, j is the number of a pixel of the vertical direction).

[0089] Next, Bfx is the variable that stores the number of pixels with respect to the horizontal direction of the images of file icons

21 ~ 23; Bfy is the variable that stores the number of pixels with respect to the vertical direction of the images of file icons **21 ~ 23**; Bf(i, j) is the array that stores the bit maps of the images of file icons **21 ~ 23** (i is the number of a pixel of the horizontal direction, j is the number of a pixel of the vertical direction).

[0090] Next, Bdx is the variable that stores the number of pixels with respect to the horizontal direction of the image of the drag icon **16**; Bdy is the variable that stores the number of pixels with respect to the vertical direction of the image of the drag icon **16**; Bd(i, j) is the array that stores the bit map of the image of the drag icon **16** (i is the number of a pixel of the horizontal direction, j is the number of a pixel of the vertical direction).

[0091] Furthermore, at the time of the activation of the present device, variable Bfx is regarded as being initialized with the number of pixels with respect to the horizontal direction of the images of file icons **21 ~ 23**; variable Bfy is regarded as being initialized with the number of pixels with respect to the vertical direction of the images of file icons **21 ~ 23**; array Bf(i, j) is regarded as being initialized with the bit maps of the images of file icons **21 ~ 23**.

[0092] Furthermore, variable Bdx is regarded as being initialized with the number of pixels with respect to the horizontal direction of the image of the drag icon **16**; variable Bdy is regarded as being initialized with the number of pixels with respect to the vertical direction of the image of the drag icon **16**; array Bd(i, j) is regarded as being initialized with the bit map of the image of the drag icon

16.

[0093] Then, these initialized variables and arrays are passed to the display controller **6** at the time of the display of each icon. Furthermore, when the shared window **12** is displayed, arrays Wx and Wy are regarded as being initialized with the position of the shared window **12** within the display screen **11**.

[0094] Fig. 14 is a drawing that shows the format of a packet of an image file sent from a circuit controller **8**. In Fig. 14, Lp is a field that stores the length of the packet; Cm is a field that stores the class of command; Fx(1) is a field that stores the value of array Sx(1)'; Fx(2) is a field that stores the value of array Sx(2)'; Fy(1) is a field that stores the value of array Sy(1)'; Fy(2) is a field that stores the value of array Sy(2)'; Bm(i, j) is a field that stores the value the bit map of the image **26** in the file displayed (i is the number of a pixel of the horizontal direction, j is the number of a pixel of the vertical direction).

[0095] Next, in Fig. 15 ~ Fig. 19, the flowcharts of the processes activated by means of the CPU 1 at the time of the occurrence of each event when image expansion is performed, among the programs stored in ROM 2, are set forth.

[0096] Fig. 15 is a flowchart of the process activated at the time of the click of a mouse. Namely, this flowchart sets forth the process when the cursor **24** of Fig. 11 (a) is caused to be displayed.

[0097] In Fig. 15, first, in step P1, whether or not the class of the object read from the display controller **6** and presently being

selected is a "shared window" is determined, and if it is a "shared window", step P2 is proceeded to. If it isn't, the process ends.

[0098] Next, the processes of steps P2 ~ P6 are, practically, the same as the processes of steps P2 ~ P6 set forth in Fig. 6. The differences are just that in the flowchart of Fig. 6, because text was handled, variables Tx, Ty of character units and arrays Sx(1), Sx(2), Sy(1), Sy(2) of character units were used, but in the present embodiment, because images are handled, variables Px, Py of pixel units and arrays Sx(1)', Sx(2)', Sy(1)', Sy(2)' of pixel units are used. Accordingly, a detailed explanation of the process content shall be omitted here.

[0099] Fig. 16 is a flowchart of the process activated at the time of commencement of the drag operation of a mouse. Namely, the process of steps P1 ~ P7 in Fig. 16 sets forth the process at the time of designating the front position of a frame **25** set forth in Fig. 11 (b), and the process of steps P8 ~ P13 sets forth the process at the time of selecting a desired file icon from among the file icons **21** ~ **23** set forth in Fig. 11 (a) and (b).

[0100] In Fig. 16, first of all, in step P1, whether or not the class of the object read from the display controller 6 and presently being selected is a "shared window" is determined, and if it is a "shared window", step P2 is proceeded to. If it isn't, step 8 is proceeded to.

[0101] In this Fig. 16 also, the same as the above-mentioned Fig. 15, the process of steps P2 ~ P14 is almost the same as the process of

steps P2 ~ P14 set forth in Fig. 7. Namely, in the present embodiment only the fact that Md' is used as the variable that stores the drag operation mode and, in addition, that variables Px, Py of pixel units and arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ of pixel units are used, differs from the flowchart of Fig. 7. Accordingly, a detailed explanation of the process contents here also shall be omitted.

[0102] Fig. 17 is a flowchart of the process activated at the time a mouse moves while dragging. Namely, the process in steps P1 ~ P6 in Fig. 17 sets forth the process during the time when the range of the frame **25** indicated in Fig. 11 (b) is designated, and the process of steps P7 ~ P11 sets forth the process when the drag icon **16** indicated in Fig. 11 (a) and (b) moves.

[0103] In Fig. 17, first, in step P1, whether or not the value of variable Md' is the value that represents "range designation" is determined, and if it is "range designation", step P2 is proceeded to. If it isn't, step P7 is proceeded to.

[0104] In step P2, from the display controller **6** the pointer **17** position information is read to variables Px and Py in pixel units. Then, in step P3, whether or not this position which the read to variables Px and Py indicate is the position, or is within the range, which the arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ indicate is determined, and, if it is in the range, step P4 is proceeded to. If it isn't, the process ends.

[0105] Furthermore, here, when the following conditions are satisfied, it is regarded to be "the position or within the range

which the arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ indicate".

[0106] When $Sx(1)' \leq Sx(2)',$ and $Sy(1)' \leq Sy(2)',$ $Wx(1) + Sx(1)' \leq Px \leq Wx(1) + Sx(2)'$ and $Wy(1) + Sy(1)' \leq Py \leq Wy(1) + Sy(2)'$
When $Sx(1)' > Sx(2)',$ and $Sy(1)' < Sy(2)',$ $Wx(1) + Sx(2)' \leq Px \leq Wx(1) + Sx(1)'$ and $Wy(1) + Sy(1)' \leq Py \leq Wy(1) + Sy(2)'$

[0107] When $Sx(1)' < Sx(2)',$ and $Sy(1)' > Sy(2)',$ $Wx(1) + Sx(1)' \leq Px \leq Wx(1) + Sx(2)'$ and $Wy(1) + Sy(2)' \leq Py \leq Wy(1) + Sy(1)'$
When $Sx(1)' > Sx(2)',$ and $Sy(1)' > Sy(2)',$ $Wx(1) + Sx(2)' \leq Px \leq Wx(1) + Sx(1)'$ and $Wy(1) + Sy(2)' \leq Py \leq Wy(1) + Sy(1)'$

[0108] Next, in step P4, the display controller **6** is instructed so as to erase the display of the frame **25** of the range which the present values of arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ indicate.
Next, in step P5, the values of variables Px and Py are set in arrays $Sx(2)', Sy(2)'$, respectively.

[0109] Then, in step P5, the display controller **6** is instructed so as to display the frame **25** in the range which the new values of arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ set as above indicate, and the process ends. Here, the frame **25** is displayed so that the line segment which the arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ make amounts to a diagonal line.

[0110] On the other hand, when step P7 is proceeded to from step P1, whether or not the value of variable Md' is the value that represents "image expansion" is determined, and if it is the value of "image expansion", step P8 is proceeded to. If it isn't, the process ends. Furthermore, since the process of step P8 ~ P11 is the same as

the process of steps P7 ~ P10 of the flowchart set forth in Fig. 8, an explanation is omitted.

[0111] Fig. 18 is a flowchart of the process activated at the time of mouse drop. In Fig. 18, first, in step P1, whether or not the value of variable Md' is the variable that represents "image expansion" is determined, and if it is the value of "image expansion", step P2 is proceeded to. If it isn't, the process ends.

[0112] In step P2 the display controller **6** is instructed so as to erase the display of the drag icon **16** of the position which the variables Dx and Dy indicate. Continuing, in step P3, the position information of the pointer **17** is read to variables Px and Py in pixel units from the display controller **6**.

[0113] Then, in step P4, whether or not the position of the pointer **17** which variables Px and Py, that have fetched this, indicate is the position, or is within the range, that arrays Sx(1)', Sx(2)', Sy(1)', Sy(2)' indicate, namely, whether or not it is the position of the cursor **24** or is within the range of the frame **25**, is determined. Then, if it is this position or is within the range, step P5 is proceeded to. If it isn't the process ends.

[0114] Furthermore, here, when the following conditions are satisfied, it is regarded as "the position of the cursor **24** or within the range of the frame **25**".

When $Sx(2) = 0$, $Sy(2) = 0$

$Px = Wx(1) + Sx(1)'$ and $Py = Wy(1) + Sx(1)'$

When $Sx(2) \neq 0$, $Sy(2) \neq 0$, and conditions the same as step P3 of Fig.

17 are satisfied

[0115] Next, in step P5, the display controller **6** is instructed so as to erase the display of the frame **25** of the range that arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ indicate. Continuing, in step p6, the number of pixels of the horizontal direction and the vertical direction of the image **26** that corresponds to file name $Fn(i)$ are read to Bx and By , respectively.

[0116] Then, in step P7, the image **26** is displayed that corresponds to the file name $Fn(i)$ at the position or in the range that arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ indicate. Here, when $Sx(2)' = 0$, $Sy(2)' = 0$, in variable $Sx(2)'$ the rather small values of $(Wx(1) + Bx)$ and $Wx(2)$ are set, in addition, in the variable $Sy(2)'$ the rather small values of $(Wy(2) + By)$ and $Wy(2)$ are set. These are so that the range **26** that displays the image represented by arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ does not protrude from the shared window **12**.

[0117] Furthermore, the image **26** indicated by the array $B(i, j)$ read from the file $Fn(i)$ is increased $|Sx(1)' - Sx(2)'| / Bx$ times in the horizontal direction, and in addition, is increased $|Sy(1)' - Sy(2)'| / By$ times in the vertical direction. Here, $|\alpha|$ represents the absolute value of the alpha formula. /11

[0118] Next, in step P8, the image expansion command is stored in field Cm , and the values of arrays $Sx(1)', Sx(2)', Sy(1)', Sy(2)'$ are stored, respectively, in field $Fx(1)$, $Fx(2)$, $Fy(1)$, $Fy(2)$, and the value of the image indicated by array $B(i, j)$ is stored in field $Bb(i,$

j). And, the information of each of these fields is sent to a partner device via a circuit controller **8**, and the process ends.

[0119] Fig. 19 is a flowchart of the process activated at the time the circuit controller **8** has received the packet containing the image expand command in the field Cm. First, in step P1, the information of the fields Cx(1), Cx(2), Cy(1), Cy(2) received from the circuit controller **8** is stored in arrays Rx(1)', Rx(2)', Ry(1)', Ry(2)', respectively. Next, in step P2, the information of field T(i) received from the circuit controller **8** is expanded in the position or range that the arrays Rx(1)', Rx(2)', Ry(1)', Ry(2)' indicate, and the process ends.

[0120] In this way, in the present embodiment, by just performing the drag operation and drop operation with respect to the file icon **21** displayed on the same display screen **11**, the image inside a file that corresponds to the above-mentioned file icon **21** can be displayed in a shared window **12** and, in addition, it can be sent to a partner device, and the operation that displays the image in the shared window **12** can be simply carried out.

[0121] Furthermore, in the above-mentioned second embodiment, the display range of the image was arranged so as to be displayed in a frame **25**, but arranging so as to display by means of attaching hatching and changing the color of the display is also acceptable.

[0122] Furthermore, in the above-mentioned second embodiment, the clicked position is made so as to be the left top edge of the image **26**, but arranging for the clicked position to be the center and the

right bottom edge and the like of the image **26** is also acceptable.

[0123] Furthermore, in step P7 of Fig. 18, there are times when the ratio of vertical and horizontal differs when expanding and contracting to align the image **26** indicated by the array $B(i, j)$ to the frame **25**, but this can be dealt with by performing the following. If done in this way, the ratio of vertical and horizontal can be maintained.

[0124] When $(|Sx(1)' - Sx(2)'| / Bx) \leq |Sy(1)' - Sy(2)'| / By$, increase $|Sx(1)' - Sx(2)'| / Bx$ in the vertical and horizontal directions respectively.

When $(|Sx(1)' - Sx(2)'| / Bx) > (|Sy(1)' - Sy(2)'| / By)$, increase $|Sy(1)' - Sy(2)'| / By$ in the vertical and horizontal directions respectively.

[0125] The above, as mentioned in the first embodiment and the second embodiment, are arranged so that, by mutually transferring a file of text and images and the like expanded or displayed inside a shared window **12** between a sending side device and a receiving side device connected via a circuit **9**, these files can be shared.

[0126] However, in conventional communications devices, when a file is transferred to a partner device during the use of the communications device, a file transfer application is separately started, and the file name and the like of the other party is designated and the transfer is carried out, or the icon of a file that one wants to transfer is dragged and dropped inside a shared window and the transfer is carried out.

[0127] However, in this kind of conventional example there was the weak point that, when a file is transferred during the use of the communications device, whether or not file reception has been completed at the receiving side device cannot be confirmed at the sending side device. The communications device based on the third embodiment mentioned next has been devised in order to resolve this kind of weak point.

[0128] Fig. 20 is a drawing that shows an example of the display of the screen of the display part 7 at the time of sending a file. In Fig. 20, 31 ~ 33 are file icons that correspond to files of text and images and the like stored in the external memory unit 4 of the sending side device. Furthermore, 34 is a file icon displayed inside a shared window 12 after the drop operation.

[0129] When one wants to send a file to a partner device, it is carried out by dragging and, after that, dropping on a shared window 12, the file icon 31 that corresponds to the file one wants to send.

[0130] Namely, first, the pointer 17 is moved onto the file icon 31 that corresponds to the file one wants to send and the drag operation commences. During this drag, the drag icon 16 is displayed at the position of the pointer 17 that moves.

[0131] After that, when the pointer 17 moves inside the shared window 12 and the drop operation is performed, the file icon 34 is displayed in the position where the drop operation was performed and, in addition, the display of the drag icon 16 disappears. And, the file that corresponds to the file icon 34 is caused to be sent to the

partner device.

[0132] At this time, the file icon **31** of the drag source is left displayed unchanged and the contents of the file that corresponds to this is maintained. After that, when the reception of the file in the partner device is finished, the display of the file icon **34** disappears. Because of this it becomes possible to confirm at the sending side whether or not the partner device has finished the reception of a file.

[0133] Fig. 21 is a drawing that shows an example of the display of the screen of the display part **7** at the time of receiving a file. In Fig. 21, **35** is the file icon displayed inside a shared window **12** when a file has been received from the sending side device. Furthermore, **36** is a file icon displayed after the drop operation. /12

[0134] When there is reception of a file, a file icon **35** is displayed in the shared window **12**. At this time, the received file is stored in a temporary file. When one wants to save the file stored in this temporary file, it is carried out by means of dragging the file icon **35** and dropping it outside of the shared window **12**.

[0135] Namely, when the pointer **17** is moved onto the file icon **35** and the drag operation starts, the drag icon **16** is displayed at the position of the pointer **17**. After that, when the pointer **17** is moved outside of the shared window **12** and the drop operation is performed, the file icon **36** is displayed at the position where the drop operation was performed and, in addition, the display of the file icon **35** and

the drag icon **16** disappears. With this, the reception of the file finishes.

[0136] Fig. 22 is a drawing that shows the memory map of the RAM 3 in the present embodiment. Furthermore, in Fig. 22, the same reference numerals as those shown in the memory map set forth in Fig. 4 are affixed and there are variables or arrays that have the same content. First, Md'' is a variable that stores the file transfer move modes ("transfer" or "receive").

[0137] Next, Cs is a variable that stores the file send request command; Cr is the variable that stores the file receive confirmation command; Sx is a variable that stores the value of variable Dx; Sy is a variable that stores the value of variable Dy.

[0138] Next, Rx is a variable that stores the value of received field Sx; Ry is a variable that stores the value of received field Sy; Fs is an array that stores the file names of files to send; Fr is an array that stores the names of the received files (values of field Fs). Furthermore, Bf is an array that stores the bit maps of file icons **31 ~ 36**; Bd is an array that stores the bit map of the drag icon 16.

[0139] Furthermore, at the time of the activation of the present device, variable Cs is regarded as being initialized with the field send request command, variable Cr is regarded as being initialized with the field receive confirmation command. Furthermore, array Bf is regarded as being initialized with the bit maps of file icons **31 ~ 36** and Bd is regarded as being initialized with the bit map of the drag

icon **16**. And the bit map information in these initialized arrays **Bf**, **Bd** is passed to the display controller **6** when each icon is displayed.

[0140] Fig. 23 is a drawing that shows the format of a packet sent from the circuit controller **8** of the sending side device. In Fig. 23, **Lp** is a field that stores the length of the packet; **Cs** is a field that stores the file send request command (variable **Cs**); **Sx** is a field that stores the variable **Sx**; **Sy** is a field that stores the variable **Sy**; **Fs** is a field that stores the file names (array **Fs**) of the files to send; **Fc** is a field that stores the contents of the files to send.

[0141] Furthermore, Fig. 24, is a drawing that shows the format of the packet sent from the circuit controller **8** of the receiving side device. In Fig. 24, **Lp** is the field that stores the length of the packet; **Cr** is the field that stores the file receive confirmation command (variable **Cr**).

[0142] Next, in Fig. 25 ~ Fig. 29 are set forth the flowcharts of the processes activated by the CPU 1 at the time of the occurrence of each event in the present embodiment among the programs stored in the ROM 2.

[0143] Fig. 25 is the flowchart of the process activated at the time of the commencement of the drag operation of a mouse. Namely, this flowchart sets forth the process at the time of designating the file icon **31** of Fig. 20 or the file icon **35** of Fig. 21.

[0144] In Fig. 25, first, in step P1, whether or not the class of the object presently selected that has been read from the display controller 6 is a "file icon" is determined, and if it is a "file

icon", step P2 is proceeded to. If it isn't, the process ends.

[0145] In step P2 the position information of the pointer **17** is read from the display controller **6** to variables Px and Py. Then, in step P3, the display controller **6** is instructed so as to display the drag icon **16** at the position which variables Px and Py that have read this indicate.

[0146] Next, in step P4, the values of variables Px and Py are stored in variables Dx and Dy, and in step P5, whether or not the position of the pointer **17** that the above-mentioned variables Px and Py indicate is inside the shared window **12** is determined. Here, when the position of the pointer **17** is not inside the shared window **12**, since it corresponds to the case in which the file icon **31** of Fig. 20 is designated, step P6 is proceeded to. On the other hand, when the position of the pointer **17** is inside the shared window **12**, since it corresponds to the case in which the file icon **35** of Fig. 21 is designated, step P8 is proceeded to.

[0147] Furthermore, here, The position information of the shared window **12** is read from the display controller **6**, and, when the position information that has been read satisfies the following conditions, it is regarded as "within a shared window".

Left edge of shared window < Px < right edge of shared window

Bottom edge of shared window < Py < top edge of shared window

[0148] Then, when step P6 is proceeded to from step P5, the value of variable Md'' is set to the value that represents "send". And, in step P7, the file name (here, "File 1") that corresponds to the file

icon **31** is read to array Fs from the display controller **6**, and the process ends. On the other hand, when step P8 is proceeded to from the above-mentioned step P5, the value of variable Md'' is set to the value that represents "receive", and the process ends.

[0149] Fig. 26 is a flowchart of the process activated at the time a mouse moves while dragging. In Fig. 26, first, in step P1, the display controller **6** is instructed so that the display of the drag icon **16** in the position that variables Dx and Dy indicate is erased. /13

[0150] Next, in step P2, the present position information of the pointer **17** is read to variables Px and Py from the display controller **6**. And, in step P3, the display controller **6** is instructed so as to display the drag icon **16** in the position that variables Px and Py that have read this indicate. Next, in step P4, the values of variables Px and Py are stored in variables Dx and Dy, respectively, and the process ends.

[0151] Fig. 27 is a flowchart of the process activated at the time of the drop of a mouse. Namely, the process of steps P1 ~ P6 in Fig. 27 sets forth the process when file icon **31** of Fig. 20 is dropped within a shared window **12** and file icon **34** is displayed. Furthermore, the process of steps P7 ~ P11 sets forth the process when file icon **35** of Fig. 21 is dropped outside of a shared window **12** and file icon **36** is displayed.

[0152] In Fig. 27, first, in step P1, the display controller **6** is instructed so as to erase the display of the drag icon **16** of the

position that variables Dx and Dy indicate. Next, in step P2, whether or not the value of variable Md'' is the value that represents "send" is determined, and if it is the value of "send", step P3 is proceeded to. If it isn't, step P7 is proceeded to.

[0153] In step P3, whether or not the position that variables Dx and Dy indicate is within the shared window **12** is determined, and if it is within the shared window **12**, step P4 is proceeded to. If it isn't, the process ends. Here, the determination of whether or not the position is within the shared window **12** is carried out with the same conditions as the determination in step P5 of Fig. 25.

[0154] When step P4 has been proceeded to from the above-mentioned step P3, the display controller **6** is instructed so as to display, in the position that variables Dx and Dy indicate, the file icon **34** that corresponds to the file name read into array Fx. Furthermore, in step P5, the values of variables Dx and Dy are stored in variables Sx and Sy, respectively.

[0155] Then, in step P6, the values of variable Cs, variable Sx, variable Sy, array Fs, as well as the contents of the files that correspond to the file names read into array Fs, are set into fields Cs, Sx, Sy, Fs, Fc, respectively, and these are sent to the circuit controller **8** and the process ends.

[0156] On the other hand, when step P7 is proceeded to from the above-mentioned step P2, whether or not the position that variables Dx and Dy indicate is within the shared window **12** is determined, and if it is within the shared window **12**, step P8 is proceeded to. If it

isn't, the process ends. The determination of whether or not the position is within a shared window **12** in this step P7 also is carried out with conditions that are the same as the determination in step P5 of Fig. 25.

[0157] In step P8, the display controller **6** is instructed so as to erase the display of the file icon **35** in the position that variables Rx and Ry indicate, and in step P9, the display controller **6** is instructed so as to display, at the position that variables Dx and Dy indicate, the file icon **36** that corresponds to the file name read into array Fs.

[0158] Next, in step P10, the file name of the temporary file is changed to the name indicated by the value of array Fs. Then, in step P11, the value of variable Cr is set in field Cr, and this is sent to the circuit controller **8** and the process ends.

[0159] Fig. 28 is a flowchart of the process activated at the time a circuit controller **8** has received a packet that contains a Fig. 23 kind of field Cs. First, in step P1, the values of fields Sx, Sy, Fs received from a circuit controller **8** are stored, respectively, in variable Rx, variable Ry and array Fr.

[0160] Next, in step P2, the value of field Fc received from a circuit controller **8** is stored in temporary file and, in addition, in step P3, the display controller **6** is instructed so as to display, at the position that variables Rx and Ry indicate, a file icon **35** that corresponds to the file name read into array Fs, and the process ends.

[0161] Fig. 29 is a flowchart of the process activated at the time a circuit controller **8** has received a packet that contains a Fig. 24 kind of field Cr. In this case, in step P1, the display controller **6** is instructed so as to erase the display of the file icon **34** of the position that variables Sx and Sy indicate, and the process ends.

[0162] Furthermore, in the above-mentioned third embodiment the case in which the shared window **12** of the sending side and the receiving side is in the same position inside the screen **11** is explained. In contrast to this, arranging so as to send to the receiving side relative position information of the file icon **34** within the shared window **12** of the sending side is also acceptable. If done in this way, it becomes unnecessary for the position of the shared window **12** to always be in the same position in the sending side and the receiving side.

[0163] Furthermore, in the above-mentioned third embodiment, file icon **34** of Fig. 20 and file icon **35** of Fig. 21 are displayed in the same position inside the shared window **12** in the sending side and the receiving side, but arranging for the file icon **35** to be displayed in a fixed position of the shared window **12** of the receiving side (for example, the upper left edge and the center and the like) is also acceptable.

[0164] At this time, variables Rx and Ry are initialized in advance with values that indicate a fixed position that displays the file icon **35**. Furthermore, fields Sx and Sy are not necessary.

Consequently, in step P6 of Fig. 27 the values of fields Sx and Sy are not sent to the receiving side device. Furthermore, in step P1 of Fig. 28 the values of fields Sx and Sy are not received, and the respective values are also not stored in variables Rx and Ry.

[0165] Furthermore, in the above-mentioned third embodiment, /14 the name of the received file becomes the same as that of the sending side, but, before dragging the file icon 35 outside of the shared window 12, arranging so that a new name of the received file is accepted is also possible.

[0166] Furthermore, in the above-mentioned first ~ third embodiments, a LAN is used as the circuit 9 connected to the terminals, but implementing with the ISDN basic interface, the ISDN primary rate interface, broadband ISDN, a circuit switching network, a packet switching network, a public telephone network, a dedicated line, a private branch exchange and the like is also possible.

[0167] Furthermore, in the above-mentioned first ~ third embodiments, ROM 2 and RAM 3 are used as the memory units that store programs and data, but implementing also with floppy disks, hard disks, IC cards and the like is possible.

[0168] Furthermore, in the above-mentioned first ~ third embodiments, it was arranged so that the dragging of file icons 13 ~ 15, the expansion of text and the display of images or the sending of a file and the like were carried out, but arranging so that a folder icon is dragged and the text inside of a folder is expanded, the images displayed and the files sent is also acceptable. Furthermore, a

folder is that which has put together multiple files and affixed one name thereto.

[0169] Furthermore, in the above-mentioned first ~ third embodiments, text and images were sent in one packet, but when the amount of data of the text and images is great, arranging so as to split into multiple packets and send is also acceptable.

[0170] [Effects of the Invention]

As for the present invention, as mentioned above, a means is provided which, by performing prescribed operations, for example, the drag operation and the drop operation, with respect to icons displayed that correspond to various files, expands the information (for example, text and images) inside the above-mentioned files in a shared screen, and shares the information inside the above-mentioned files with a partner side device, and so, just by performing the drag operation and drop operation with respect to the above-mentioned icons without separately activating an editor, the information that corresponds to the above-mentioned files can be taken into a shared screen and the operation of taking information inside files into a shared screen can be easily and quickly carried out.

[0171] According to another characteristic of the present invention, when a file is sent from the sending side device to the receiving side device, the file icon that corresponds to that sent file is displayed in the sending side device, and when file reception is finished in the receiving side device, a means is provided that clears the display of the above-mentioned file icon and so, when

sending a file to the partner side device while the communications device is being used, whether or not file reception is finished at the receiving side device can be confirmed at the sending side and smooth information exchange can be carried out.

[BRIEF DESCRIPTION OF THE DRAWINGS]

[Fig. 1] is block diagram that shows one embodiment of the communications device of the present invention.

[Fig. 2] is a drawing that shows an example of the display of the display screen before text file expansion.

[Fig. 3] is a drawing that shows an example of the display of the display screen after text file expansion.

[Fig. 4] is a drawing that shows the memory map of RAM.

[Fig. 5] is a drawing that shows the format of a packet sent from the circuit controller.

[Fig. 6] is a flowchart of the process activated at the time of the click of a mouse.

[Fig. 7] is a flowchart of the process activated at the time of commencement of the drag operation of a mouse.

[Fig. 8] is a flowchart of the process activated at the time a mouse moves while dragging.

[Fig. 9] is a flowchart of the process activated at the time of the drop operation of a mouse.

[Fig. 10] is a flowchart of the process activated at the time the packet containing the text expand command has been received.

[Fig. 11] is the drawing that shows an example of the display of the display screen before the expansion of an image file.

[Fig. 12] is a drawing that shows an example of the display of the display screen after the expansion of an image file.

[Fig. 13] is a drawing that shows the memory map of RAM.

[Fig. 14] is a drawing that shows the format of a packet sent from the circuit controller.

[Fig. 15] is a flowchart of the process activated at the time of the click of a mouse.

[Fig. 16] is a flowchart of the process activated at the time of commencement of the drag operation of a mouse.

[Fig. 17] is a flowchart of the process activated at the time a mouse moves while dragging.

[Fig. 18] is a flowchart of the process activated at the time of the drop operation of a mouse.

[Fig. 19] is a flowchart of the process activated at the time the packet containing the image expand command has been received.

[Fig. 20] is a drawing that shows an example of the display of a display screen at the time of sending a file.

[Fig. 21] is a drawing that shows an example of the display of the display screen at the time of receiving a file.

[Fig. 22] is a drawing that shows the memory map of RAM.

[Fig. 23] is a drawing that shows the format of a packet sent from the circuit controller.

[Fig. 24] is a drawing that shows another format of a packet sent from the circuit controller.

[Fig. 25] is a flowchart of the process activated at the time of the commencement of the drag operation of a mouse.

[Fig. 26] is a flowchart of the process activated at the time a mouse moves while dragging.

[Fig. 27] is a flowchart of the process activated at the time of the drop operation of a mouse.

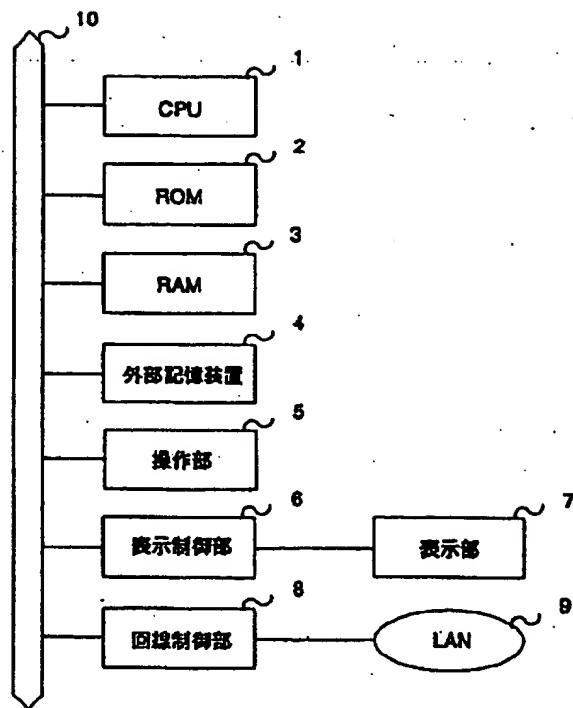
[Fig. 28] is a flowchart of the process activated at the time the packet of Fig. 23 has been received.

[Fig. 29] is a flowchart of the process activated at the time the packet of Fig. 24 has been received.

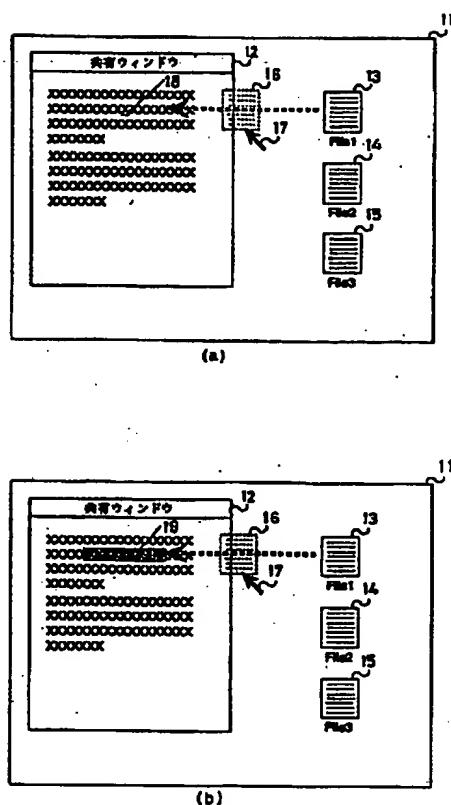
[EXPLANATION OF THE REFERENCE NUMERALS]

- 1 CPU
- 2 ROM
- 3 RAM
- 4 External memory unit
- 5 Final control element
- 6 Display controller
- 7 Display part
- 8 Circuit controller
- 9 Circuit
- 10 Bus

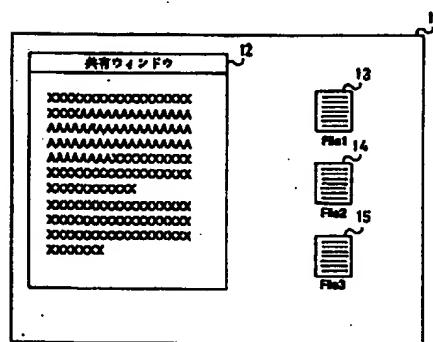
[Figure 1]



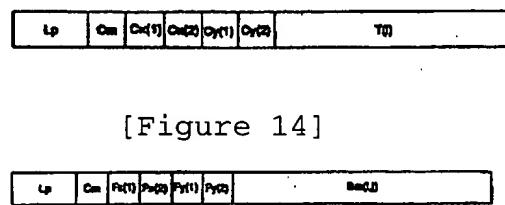
[Figure 2]



[Figure 3]



[Figure 5]



[Figure 14]

[Fig. 1]

1 CPU
2 ROM
3 RAM
4 External memory unit
5 Final control element
6 Display controller
7 Display part
8 Circuit controller
9 LAN

[Fig. 2]

12 Shared window

(a)

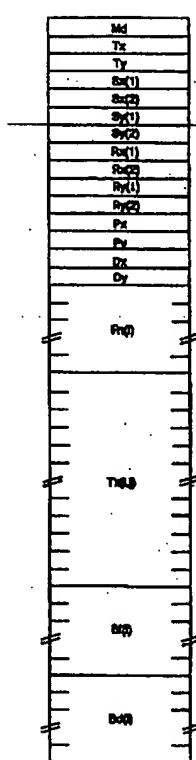
12 Shared window

(b)

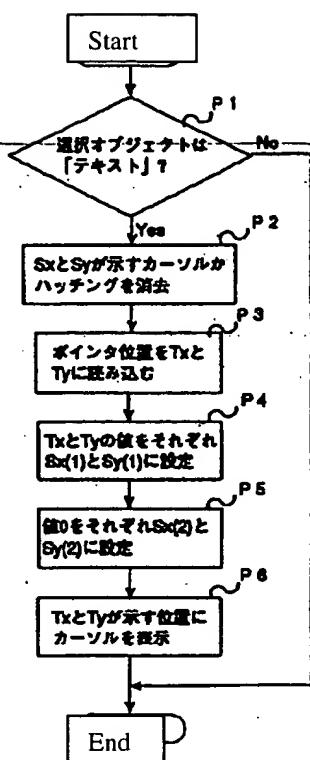
[Fig. 3]

12 Shared window

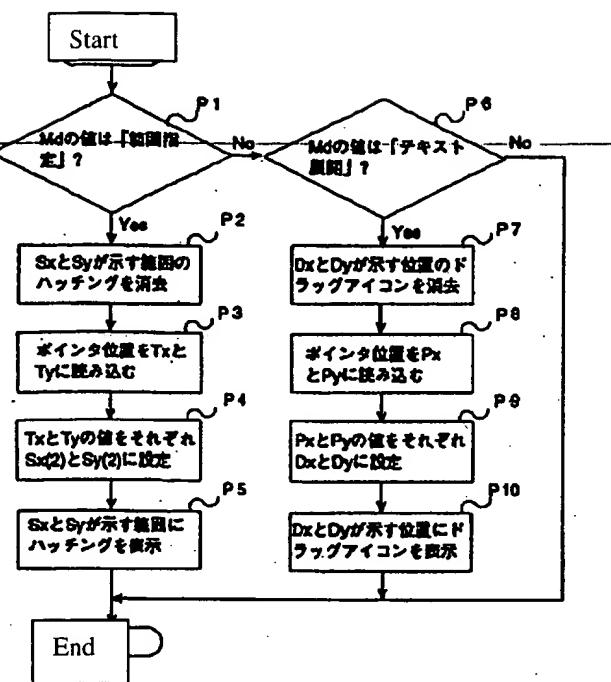
[Figure 4]



[Figure 6]



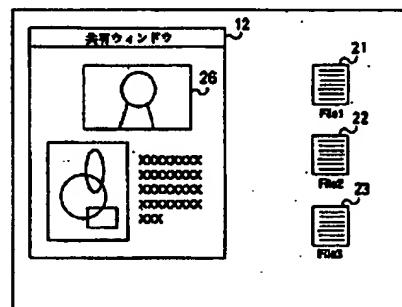
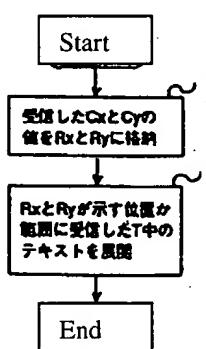
[Figure 8]



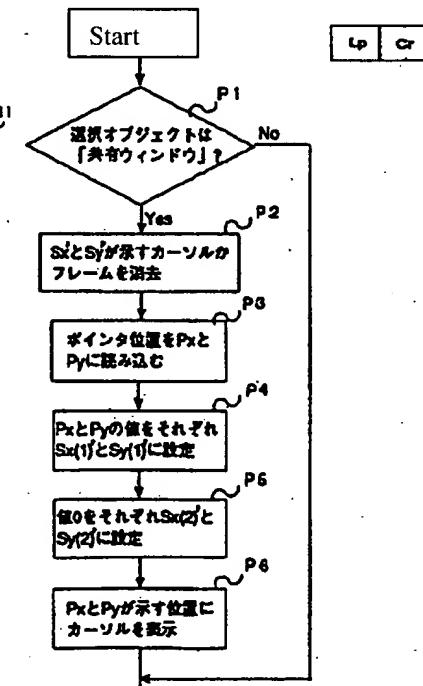
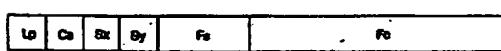
[Figure 24]

[Figure 12]

[Figure 10]



[Figure 23]



[Fig. 6]

```
Start
P1  Is selected object "text"?
P2  Erase cursor or hatching that Sx and Sy indicate
P3  Read pointer position to variables Tx and Ty
P4  Set values of Tx and Ty in Sx(1) and Sy(1) respectively
P5  Set value 0 in Sx(2) and Sy(2) respectively
P6  Display cursor at position Tx and Ty indicate
End
```

[Fig. 8]

```
Start
P1  Is the value of Md "range designation"?
P2  Erase hatching of the range that Sx and Sy indicate
P3  Read the pointer position to Tx and Ty
P4  Set the value of Tx and Ty in Sx(2) and Sy(2) respectively
P5  Display hatching in the range that Sx and Sy indicate
P6  Is the value of Md "text expansion"?
P7  Erase the drag icon of the position that Dx and Dy indicate
P8  Read the pointer position to Px and Py
P9  Set the values of Px and Py in Dx and Dy respectively
P10 Display the drag icon in the position that Dx and Dy indicate
End
```

[Fig. 10]

```
Start
P1  Store the values of Cx and Cy that have been received in Rx and
Ry
P2  Display the text in T that has been received in the position or
range that Rx and Ry indicate
End
```

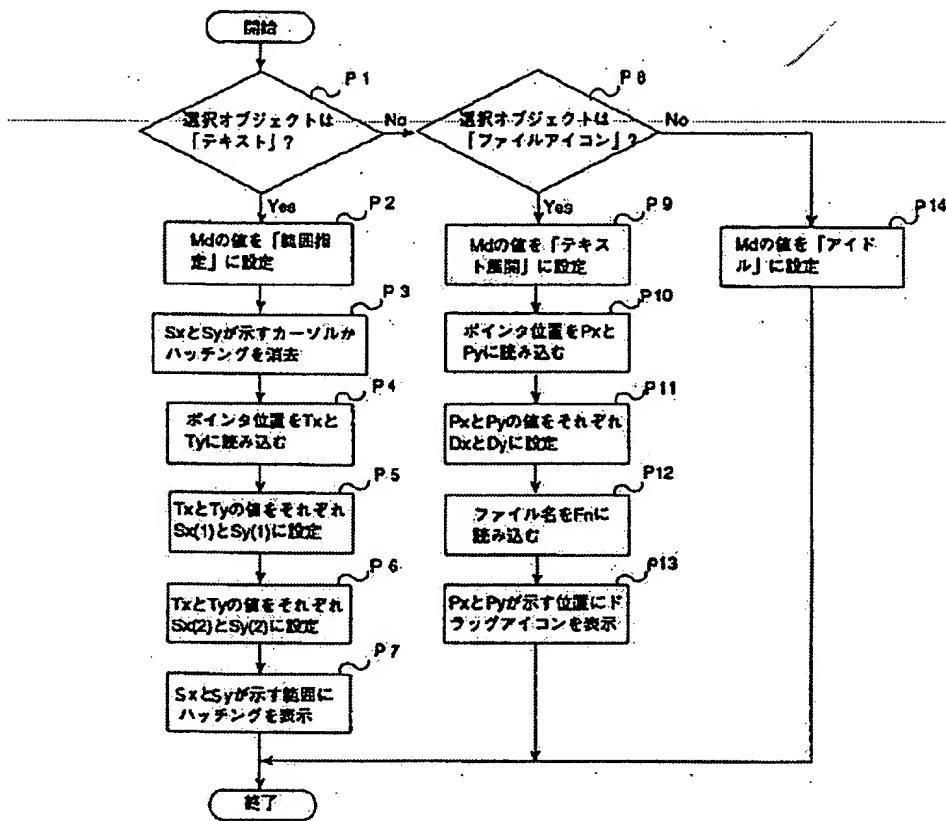
[Fig. 12]

```
12  Shared window
```

[Fig. 15]

```
Start
P1  Is the selected object a "shared window"?
P2  Erase the cursor or frame that Sx' and Sy' indicate
P3  Read the pointer position to Px and Py
P4  Set the values of Px and Py in Sx(1)' and Sy(1)' respectively
P5  Set the value 0 in Sx(2)' and Sy(2)' respectively
P6  Display the cursor in the position that Px and Py indicate
End
```

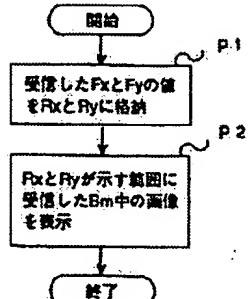
[Figure 7]



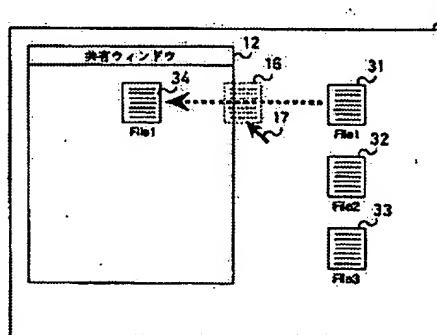
[Figure 13]

Md ⁿ	
Px	
Py	
Wx(1)	
Wx(2)	
Wy(1)	
Wy(2)	
Sx(1)	
Sx(2)	
Sy(1)	
Sy(2)	
Rx(1)	
Rx(2)	
Fx(1)	
Fx(2)	
Bx	
By	
Fn	
Bm	
Bm(1)	
Bm(2)	
Bm(3)	
Bm(4)	

[Figure 19]



[Figure 20]



[Figure 22]

Md ⁿ	
Px	
Py	
Dx	
Dy	
Cx	
Cy	
Sx	
Sy	
Rx	
Ry	
Pz	
Fz	
Gz	
Bz	

[Fig. 7]

Start

- P1 Is the selected object "text"?
- P2 Set the value of Md to "range designation"
- P3 Erase the cursor or hatching that Sx and Sy indicate
- P4 Read the pointer position to Tx and Ty
- P5 Set the values of Tx and Ty in Sx(1) and Sy(1) respectively
- P6 Set the values of Tx and Ty in Sx(2) and Sy(2) respectively
- P7 Display hatching in the range that Sx and Sy indicate
- P8 Is the selected object a "file icon"?
- P9 Set the value of Md to "text expansion"
- P10 Read the pointer position to Px and Py
- P11 Set the values of Px and Py in Dx and Dy respectively
- P12 Read the file name to Fn
- P13 Display the drag icon in the position that Px and Py indicate
- P14 Set the value of Md to "idle"

End

[Fig. 19]

Start

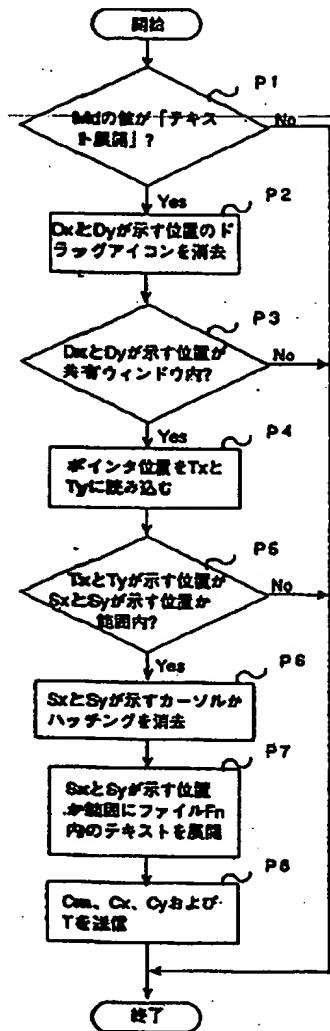
- P1 The received Fx and Fy values are stored in Rx and Ry
- P2 The image in the received Bm is displayed in the range that Rx and Ry indicate

End

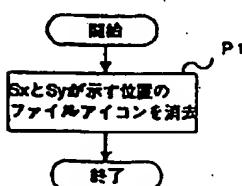
[Fig. 20]

12 Shared window

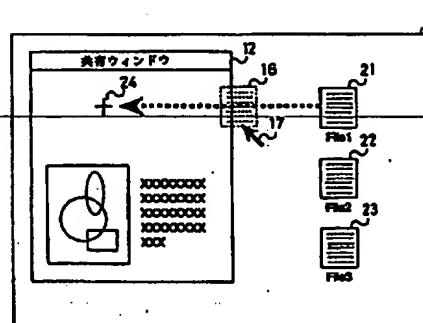
[Figure 9]



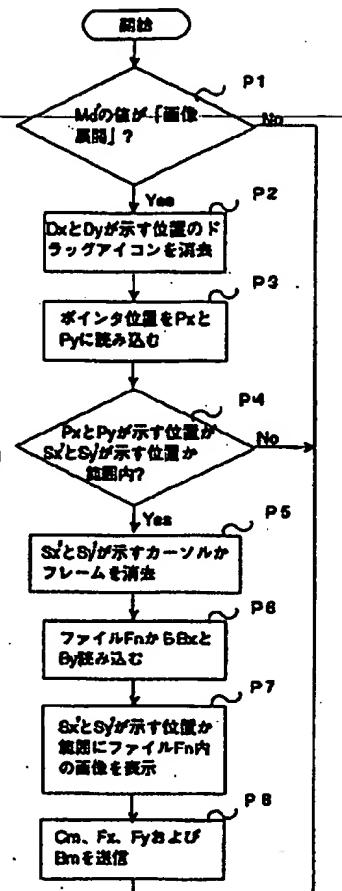
[Figure 29]



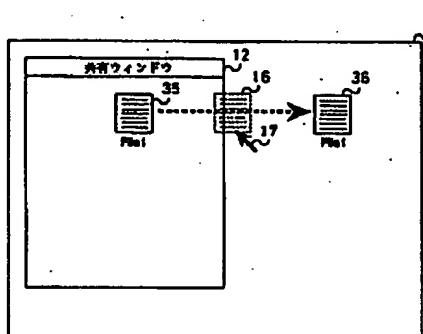
[Figure 11]



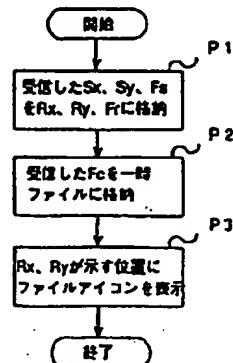
[Figure 18]



[Figure 21]



[Figure 28]



[Fig. 9]

Start

- P1 Is the value of Md "text expansion"?
- P2 Erase the drag icon of the position that Dx and Dy indicate
- P3 Is the position that Dx and Dy indicate within the shared window?
- P4 Read the pointer position to Tx and Ty
- P5 Is the position that Tx and Ty indicate the position or within the range that Sx and Sy indicate?
- P6 Erase the cursor or hatching that Sx and Sy indicate
- P7 Expand the text within file Fn in the position or range that Sx and Sy indicate
- P8 Send Cm, Cx, Cy and T

End

[Fig. 11]

- 12 Shared window
- (a)
- 12 Shared window
- (b)

[Fig. 18]

Start

- P1 Is the value of Md' "image expansion"?
- P2 Erase the drag icon of the position that Dx and Dy indicate
- P3 Read the pointer position to Px and Py
- P4 Is the position that Px and Py indicate the position, or within the range, that Sx' and Sy' indicate?
- P5 Erase the cursor or frame that Sx' and Sy' indicate
- P6 Read to Bx and By from file name Fn
- P7 Display the image in the file Fn at the position or in the range that Sx' and Sy' indicate
- P8 Send Cm, Fx, Fy and Bm

End

[Fig. 29]

Start

- P1 Erase the file icon of the position that Sx and Sy indicate
- End

[Fig. 21]

- 12 Shared window

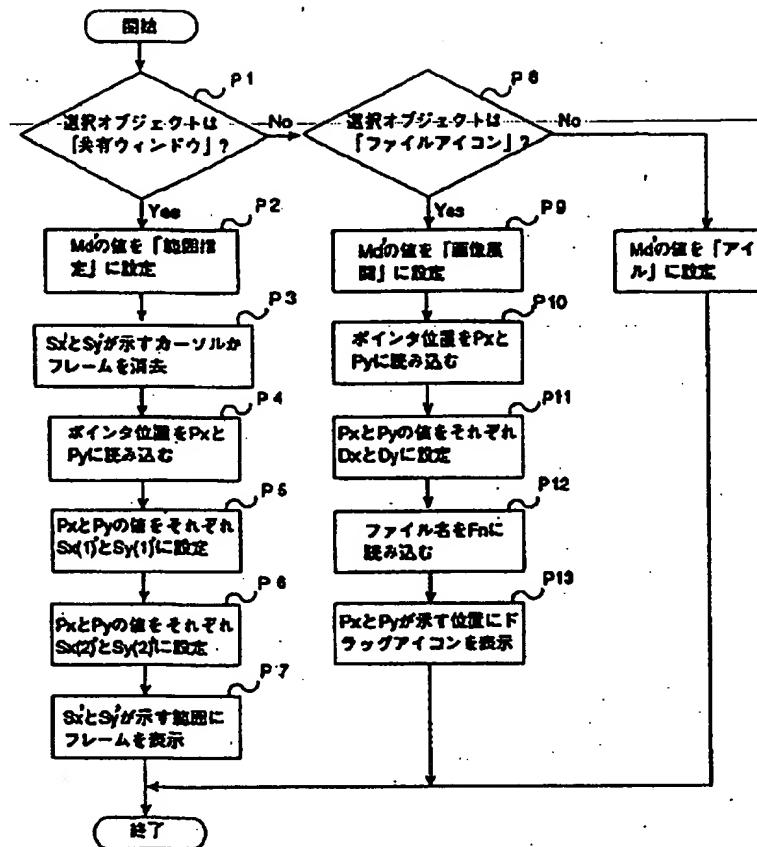
[Fig. 28]

Start

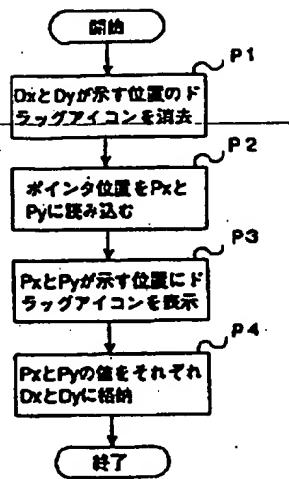
- P1 Store received Sx, Sy, Fs in Rx, Ry, Fr
- P2 Store received Fc in a temporary file
- P3 Display a file icon at the position that Rx, Ry indicate

End

[Figure 16]



[Figure 26]



[Fig. 16]

Start

- P1 Is the selected object a "shared window"?
- P2 Set the value of Md to "range designation"
- P3 Erase the cursor or frame that Sx' and Sy' indicate
- P4 Read the pointer position to Px and Py
- P5 Set the values of Px and Py to Sx(1)' and Sy(1)' respectively
- P6 Set the values of Px and Py to Sx(2)' and Sy(2)' respectively
- P7 Display a frame in the range that Sx' and Sy' indicate
- P8 Is the selected object a "file icon"?
- P9 Set the value of Md' to "image expansion"
- P10 Read the pointer position to Px and Py
- P11 Set the values of Px and Py in Dx and Dy respectively
- P12 Read the file name to Fn
- P13 Display a drag icon in the position that Px and Py indicate
- P14 Set the value of Md' to "idle"

End

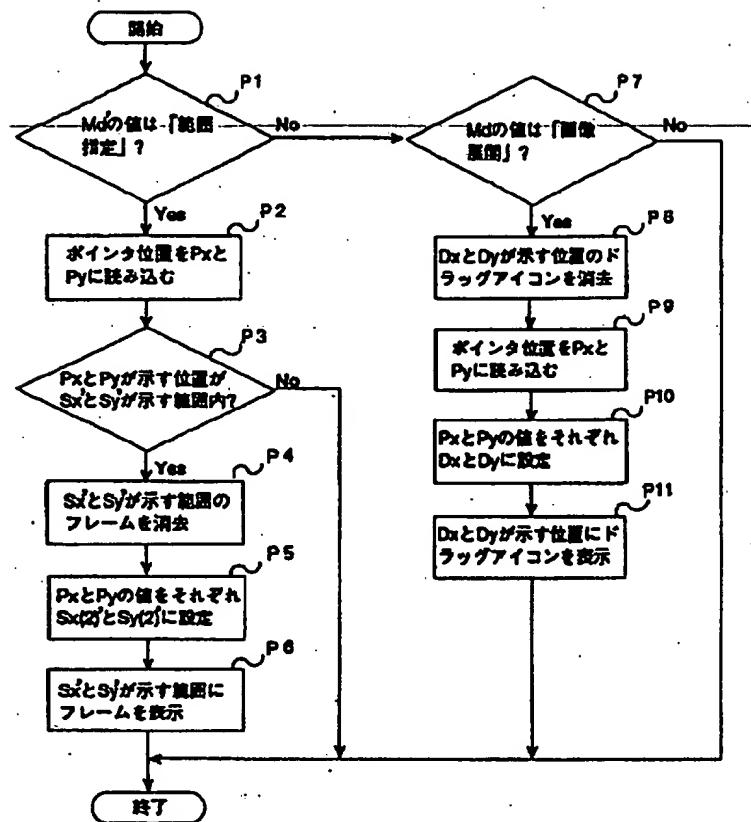
[Fig. 26]

Start

- P1 Erase the drag icon of the position that Dx and Dy indicate
- P2 Read the pointer position to Px and Py
- P3 Display the drag icon in the position that Px and Py indicate
- P4 Store the values of Px and Py in Dx and Dy respectively

End

[Figure 17]

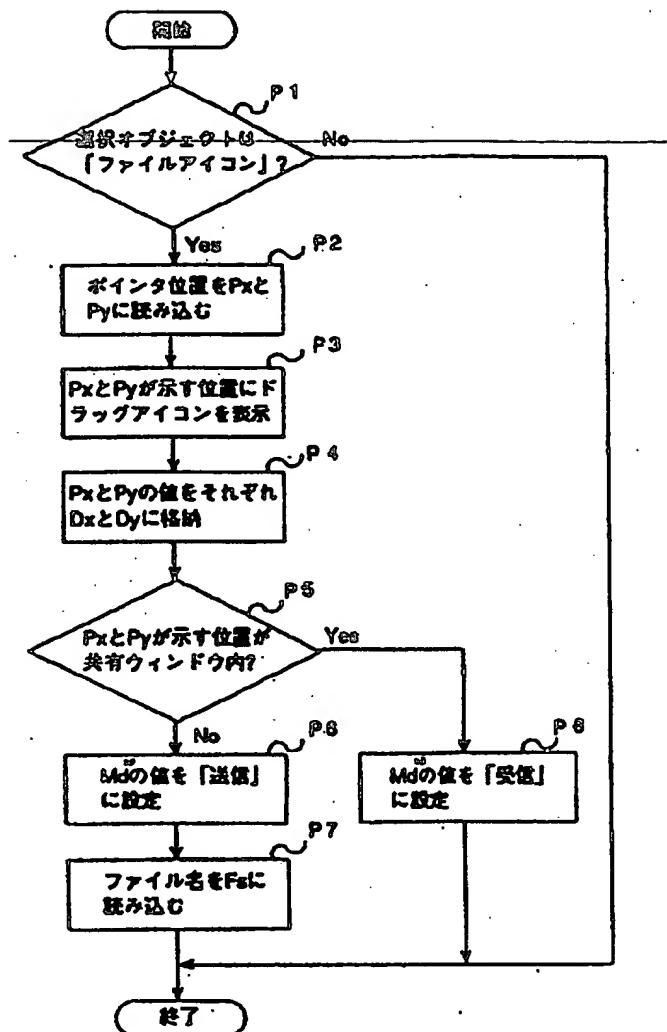


[Fig. 17]

Start

- P1 Is the value of Md' "range designation"?
- P2 Read the pointer position to Px and Py
- P3 Is the position that Px and Py indicate within the range that Sx' and Sy' indicate?
- P4 Erase the frame of the range that Sx' and Sy' indicate
- P5 Set the values of Px and Py in Sx(2)' and Sy(2)' respectively
- P6 Display a frame in the range that Sx' and Sy' indicate
- P7 Is the value of Md "image expansion"?
- P8 Erase the drag icon of the position that Dx and Dy indicate
- P9 Read the pointer position to Px and Py
- P10 Set the values of Px and Py in Dx and Dy respectively
- P11 Display the drag icon in the position that Dx and Dy indicate
- End

[Figure 25]

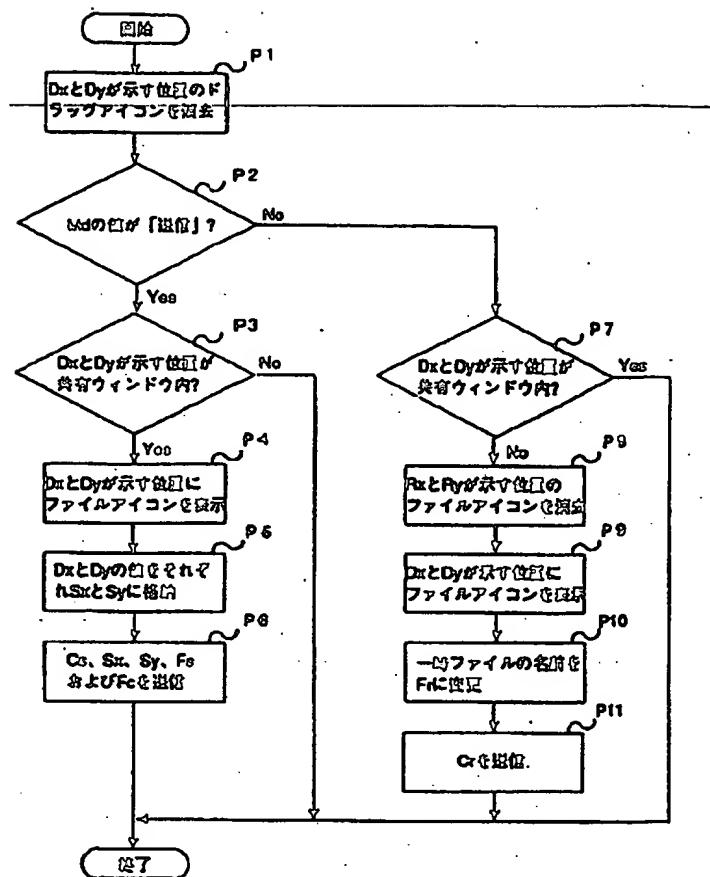


[Fig. 25]

Start

- P1 Is the selected object a "file icon"?
 - P2 Read the pointer position to Px and Py
 - P3 Display the drag icon at the position that Px and Py indicate
 - P4 Store the values of Px and Py in Dx and Dy respectively
 - P5 Is the position that Px and Py indicate inside the shared window?
 - P6 Set the value of Md to "send"
 - P7 Read the file name to Fs
 - P8 Set the value of Md to "receive"
- End

[Figure 27]



[Fig. 27]

Start

- P1 Erase the drag icon of the position that Dx and Dy indicate
 - P2 Is the value of Md "send"?
 - P3 Is the position that Dx and Dy indicate within the shared window?
 - P4 Display a file icon in the position that Dx and Dy indicate
 - P5 Store the values of Dx and Dy in Sx and Sy respectively
 - P6 Send Cs, Sx, Sy, Fs and Fc
 - P7 Is the position that Dx and Dy indicate within the shared window?
 - P8 Erase the file icon of the position that Rx and Ry indicate
 - P9 Display a file icon in the position that Dx and Dy indicate
 - P10 Change the name of the temporary file to Fr
 - P11 Send Cr
- End

特開平7-336659

(43) 公開日 平成7年(1995)12月22日

(51) Int. C1.⁶

識別記号

庁内整理番号

F I

技術表示箇所

H 04 N 7/15

G 06 F 3/14 3 7 0 A

13/00 3 5 4 D 7368-5.E

15/00 3 1 0 R 9364-5 L

審査請求 未請求 請求項の数 17 O L

(全 22 頁)

(21) 出願番号 特願平6-125505

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(22) 出願日 平成6年(1994)6月7日

(72) 発明者 門脇 修一

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(74) 代理人 弁理士 國分 孝悦

PTO 2004-0086

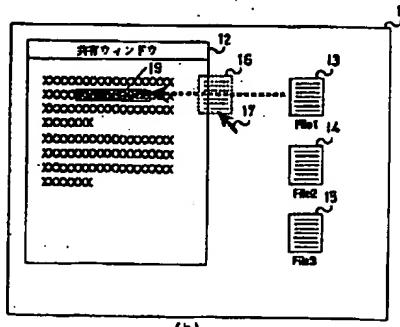
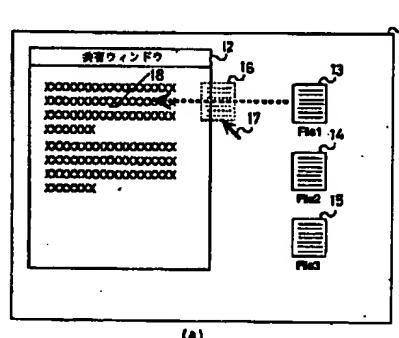
S.T.I.C. Translations Branch

(54) 【発明の名称】通信装置

(57) 【要約】

【目的】 ファイル内の情報を共有ウィンドウに書き込む操作を簡便かつ迅速に行えるようにする。

【構成】 種々のファイルに対応して表示されているファイルアイコン①③～⑤をドラッグし、これを共有ウィンドウ②内に所望の位置にドロップすることにより、上記ファイル内の情報（例えばテキストや画像）を共有ウィンドウ②内に展開するとともに、回線を介して接続された相手側の通信装置に対して上記ファイル内の情報を送信してファイル内の情報を共有するようすることにより、エディタを一々起動しなくても、上記アイコンに対してドラッグ操作およびドロップ操作をするだけで、上記アイコンに対応する情報を共有ウィンドウ②に取り込むことができるようにして、共有ウィンドウ②にファイル内の情報を取り込む操作を簡便かつ迅速に行うことができるようとする。



【特許請求の範囲】

【請求項1】 回線を介して相手側の装置と接続して用いられ、上記相手側装置と共有画面を相互に有するようになされた通信装置において、上記通信装置に格納されているファイルに対応して表示されているアイコンに対して行われる操作に応じて上記ファイル内の情報を上記共有画面に展開して上記相手側装置と共有するようにする手段を設けたことを特徴とする通信装置。

【請求項2】 上記操作は、アイコンに対するドラッグ操作およびドロップ操作であることを特徴とする請求項1に記載の通信装置。

【請求項3】 回線を介して相手側の装置と接続して用いられ、上記相手側装置と共有画面を相互に有するようになされた通信装置において、

上記通信装置の共有ウィンドウ内において所望の位置をクリック操作により指定するクリック手段と、

上記共有ウィンドウの外に表示されているファイルアイコンを上記共有ウィンドウの外から内にドラッグするドラッグ手段と、

上記ドラッグ手段によりドラッグされたファイルアイコンを、上記クリック手段により指定された共有ウィンドウ内の位置にドロップするドロップ手段と、

上記ドロップ手段によるドロップ操作の後に、上記ドラッグ手段によりドラッグされたファイルアイコンに対応するファイル内の情報を、上記クリック手段により指定された位置に挿入する挿入手段と、

上記クリック手段により指定された位置の情報および上記ドラッグ手段によりドラッグされたファイルアイコンに対応するファイル内の情報を相手側装置に送信する送信手段とを設けたことを特徴とする通信装置。

【請求項4】 請求項3に記載の通信装置において、上記相手側装置の共有ウィンドウ内における指定位置の情報および相手側装置のファイル内の情報を受信する受信手段と、

上記受信手段で受信した自装置側の共有ウィンドウ内の受信した指定位置に、上記受信したファイル内の情報を挿入する挿入手段とを設けたことを特徴とする通信装置。

【請求項5】 上記相手側装置と共有されるファイル内の情報は、テキストデータであることを特徴とする請求項1、3または4記載の通信装置。

【請求項6】 上記相手側装置と共有されるファイル内の情報は、画像データであることを特徴とする請求項1、3または4記載の通信装置。

【請求項7】 回線を介して相手側の装置と接続して用いられ、上記相手側装置と共有画面を相互に有するようになされた通信装置において、

上記通信装置の共有ウィンドウ内において所望の範囲をドラッグ操作により指定する第1のドラッグ手段と、

上記共有ウィンドウの外に表示されているファイルアイコンを上記共有ウィンドウの外から内にドラッグする第2のドラッグ手段と、
上記第2のドラッグ手段によりドラッグされたファイルアイコンを、上記第1のドラッグ手段により指定された共有ウィンドウ内の範囲にドロップするドロップ手段と、

上記ドロップ手段によるドロップ操作の後に、上記第2のドラッグ手段によりドラッグされたファイルアイコンに対応するファイル内のテキストでもって上記第1のドラッグ手段により指定された範囲のテキストを置換する置換手段と、

上記第1のドラッグ手段により指定された範囲の情報および上記第2のドラッグ手段によりドラッグされたファイルアイコンに対応するファイル内のテキストを相手側装置に送信する送信手段とを設けたことを特徴とする通信装置。

【請求項8】 請求項7に記載の通信装置において、上記相手側装置の共有ウィンドウ内における指定範囲の情報および相手側装置のファイル内のテキストを受信する受信手段と、

上記受信手段で受信した指定範囲の情報が示す自装置側の共有ウィンドウ内の範囲のテキストを、上記受信手段で受信したファイル内のテキストでもって置換する置換手段とを設けたことを特徴とする通信装置。

【請求項9】 回線を介して相手側の装置と接続して用いられ、上記相手側装置と共有画面を相互に有するようになされた通信装置において、

上記通信装置の共有ウィンドウ内において所望の範囲を

ドラッグ操作により指定する第1のドラッグ手段と、上記共有ウィンドウの外に表示されているファイルアイコンを上記共有ウィンドウの外から内にドラッグする第2のドラッグ手段と、

上記第2のドラッグ手段によりドラッグされたファイルアイコンを、上記第1のドラッグ手段により指定された共有ウィンドウ内の範囲にドロップするドロップ手段と、

上記ドロップ手段によるドロップ操作の後に、上記第2のドラッグ手段によりドラッグされたファイルアイコン

に対応するファイル内の画像を、上記第1のドラッグ手段により指定された範囲に表示する表示手段と、

上記第1のドラッグ手段により指定された範囲の情報および上記第2のドラッグ手段によりドラッグされたファイルアイコンに対応するファイル内の画像を相手側装置に送信する送信手段とを設けたことを特徴とする通信装置。

【請求項10】 請求項9に記載の通信装置において、上記相手側装置の共有ウィンドウ内における指定範囲の情報および相手側装置のファイル内の画像を受信する受信手段と、

上記受信手段で受信した指定範囲の情報が示す自装置側の共有ウィンドウ内の範囲に、上記受信手段で受信したファイル内の画像を表示する表示手段とを設けたことを特徴とする通信装置。

【請求項11】 請求項6または9に記載の通信装置において、

上記ドラッグされたファイルアイコンに対応するファイル内の画像を共有ウィンドウ内の指定範囲に表示するとき、その画像が上記指定範囲からはみ出さないように縦横同じ、または別々の倍率で拡大または縮小する手段を設けたことを特徴とする通信装置。

【請求項12】 上記ファイルアイコンのドラッグ操作中に、上記ファイルアイコンをドラッグアイコンに変更して表示する表示制御手段を設けたことを特徴とする請求項5、6、7または9に記載の通信装置。

【請求項13】 回線を介して相手側の装置と接続して用いられ、上記相手側装置と共有画面を相互に有するようになされた通信装置において、

上記通信装置に格納されているファイルに対応して表示されているファイルアイコンに対してドラッグ操作およびドロップ操作をすることにより、上記ファイルアイコンに対応するファイルを相手側装置に送信する送信手段と、

上記送信手段によるファイル送信中に、共有ウィンドウ内に上記ファイルアイコンを表示する表示手段と、上記送信手段により送信されるファイルの受信を完了したときに、上記共有ウィンドウ内に表示されているファイルアイコンを消去する表示消去手段とを設けたことを特徴とする通信装置。

【請求項14】 回線を介して相手側の装置と接続して用いられ、上記相手側装置と共有画面を相互に有するようになされた通信装置において、

上記共有ウィンドウの外に表示されているファイルアイコンを上記共有ウィンドウの外から内にドラッグするドラッグ手段と、

上記ドラッグ手段によりドラッグされたファイルアイコンを共有ウィンドウ内にドロップするドロップ手段と、上記ドロップ手段によるドロップ操作の後に、上記ドラッグ手段によりドラッグされたファイルアイコンを共有ウィンドウ内に表示する表示手段と、

上記共有ウィンドウ内に表示されたファイルアイコンに対応するファイルを相手側装置に送信する送信手段と、上記送信手段により送信されるファイルの受信を完了したときに、上記共有ウィンドウ内に表示されているファイルアイコンを消去する表示消去手段とを設けたことを特徴とする通信装置。

【請求項15】 請求項13または14に記載の通信装置において、

上記送信されてきたファイルを受信したときに、上記受信ファイルに対応するファイルアイコンを上記共有ウ

ンドウ内に表示する表示手段と、
上記表示されたファイルアイコンを上記共有ウィンドウの外にドラッグするドラッグ手段と、
上記ドラッグ手段によりドラッグされたファイルアイコンを上記共有ウィンドウの外でドロップするドロップ手段と、

上記ドロップ手段によるドロップ操作の後に、上記ファイルアイコンに対応するファイルを所定の記憶手段に保存する保存手段と、

10 上記ファイル受信が完了したことを相手側装置に通知する通知手段とを設けたことを特徴とする通信装置。

【請求項16】 請求項15に記載の通信装置において、

上記表示手段は、上記共有ウィンドウ内において相手側装置の共有ウィンドウ内に表示されているファイルアイコンと同じ位置に、上記受信ファイルに対応するファイルアイコンを表示するようになされていることを特徴とする通信装置。

【請求項17】 請求項15に記載の通信装置において、

20 上記表示手段は、上記共有ウィンドウ内の固定された位置に、上記受信ファイルに対応するファイルアイコンを表示するようになされていることを特徴とする通信装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、共有ウィンドウ上に表示されたテキストや画像等からなる会議情報を通信回線を介して相手と共有するようになされた通信装置に関する。

30 【0002】

【従来の技術】従来、通信装置は、通信回線を介して通信相手の装置に接続されて用いられる。そして、この通信装置の表示部には、共有ウィンドウが表示される。この共有ウィンドウ上には、テキストや画像等からなる会議情報が通信相手の装置と同じように表示される。

【0003】また、共有ウィンドウ内にテキストや画像等を書き込むこともでき、この変更結果は相手の共有ウィンドウに即時に反映されるようになっている。通信装置の使用中に、あるファイル内のテキストや画像を共有ウィンドウに書き込むときは、一旦エディタ等を起動してファイルを開き、そのファイル中のテキストや画像の全体を選択してコピー・アンド・ペーストするようにしている。

【0004】ここで、コピー・アンド・ペーストとは、ドラッグ操作等により選択したテキストや画像をクリップ・ボードと呼ばれる一時的な記憶領域にコピーし、クリップ・ボードから別のウィンドウ内に貼り付ける操作である。また、ドラッグ操作とは、マウスのボタンを押したままマウスを移動する操作である。

【0005】

【発明が解決しようとする課題】しかしながら、上記従来例では、通信装置の使用中にあるファイル内のテキストや画像を共有ウィンドウに書き込むときは、一旦エディタ等を起動してファイルを開き、そのファイル中のテキストや画像の全体を選択してコピー・アンド・ペーストしなければならないので、操作が煩雑であるという欠点があった。

【0006】本発明は上述の問題点にかんがみ、ファイル内の情報を共有ウィンドウに書き込む操作を簡便に、かつ迅速に行えるようにすることを目的とする。

【0007】

【課題を解決するための手段】本発明の通信装置は、回線を介して相手側の装置と接続して用いられ、上記相手側装置と共有画面を相互に有するよう成了された通信装置において、上記通信装置に格納されているファイルに対応して表示されているアイコンに対して行われる操作に応じて上記ファイル内の情報を上記共有画面に展開して上記相手側装置と共有するようにする手段を設けていく。

【0008】本発明の他の特徴とするところは、回線を介して相手側の装置と接続して用いられ、上記相手側装置と共有画面を相互に有するよう成了された通信装置において、上記通信装置に格納されているファイルに対応して表示されているファイルアイコンに対してドラッグ操作およびドロップ操作をすることにより、上記ファイルアイコンに対応するファイルを相手側装置に送信する送信手段と、上記送信手段によるファイル送信中に、共有ウィンドウ内に上記ファイルアイコンを表示する表示手段と、上記送信手段により送信されるファイルの受信を完了したときに、上記共有ウィンドウ内に表示されているファイルアイコンを消去する表示消去手段とを設けたものである。

【0009】

【作用】本発明は上記技術手段より成るので、同じ画面上に表示されたファイルアイコンに対して簡単な所定の操作をするだけで、上記ファイルアイコンに対応するファイル内の情報を共有画面（共有ウィンドウ）に展開することが可能となり、共有ウィンドウにテキストを書き込んだり画像を挿入したりする際に、エディタを一々起動しなくても済むようになる。

【0010】また、本発明の他の特徴によれば、送信側装置から受信側装置にファイルを送信すると、その送信ファイルに対応するファイルアイコンが送信側装置で表示され、その後、受信側装置でファイル受信が完了すると、上記ファイルアイコンの表示が消されるようになるので、ファイルの受信が完了したかどうかを送信側において確認することが可能となる。

【0011】

【実施例】以下、本発明の通信装置の一実施例を図面を

参照して説明する。まず、第1の実施例として、ファイル内のテキストを共有ウィンドウに展開する場合について説明する。

【0012】図1は、本実施例の通信装置の構成を示すブロック図である。図1において、1は本装置全体を制御するCPU、2は上記CPU1で実行されるプログラムを格納するROM、3は上記CPU1で使用される種々のデータを格納するRAMである。

【0013】次いで、4は上記CPU1で使用される種々のファイル（テキストや画像などのファイル）を格納する磁気ディスク等からなる外部記憶装置、5は本装置の操作を行うためのキーボードやマウス等からなる操作部、6はCRT（ブラウン管）やLCD（液晶表示ディスプレイ）等からなる表示部7を制御する表示制御部である。

【0014】次いで、8はLAN（Local Area Network）等のインターフェースを制御する回線制御部、9は本装置が通信相手の装置と接続されるLAN等の回線、10は上記の各構成ブロック1～6、および8を接続するバスである。以上のような構成の通信装置が、同様にして構成されている他の通信装置に回線9を介して接続される。

【0015】図2は、テキストファイルの展開前における表示部7の画面の表示例を示す図である。図2において、11は表示部7の画面全体、12は相手装置との間で表示されたテキストを共有する共有ウィンドウ（共有画面）である。

【0016】次いで、13～15はファイルアイコンであり、図1の外部記憶装置4に格納されている種々のテキストファイルに対応するものである。16はドラッグアイコンであり、ドラッグ操作（マウスボタンを押したままマウスを移動すること）が行われている間、表示される。

【0017】次いで、17はポインタであり、マウスやキーボード等の操作部5の操作に連動して表示画面11上を任意に移動する。このポインタ17は、操作部5を用いて種々の操作をするときに、その操作位置を指示するものである。18はクリック操作（マウスボタンを短い時間押すこと）により表示されたカーソル、19はドラッグ操作により共有ウィンドウ12内に表示されたハッチングである。

【0018】図2(a)に示すように、外部記憶装置4に格納されているファイル中のテキストを共有ウィンドウ12内に展開したいときは、まず、共有ウィンドウ12内でテキストを展開したい位置にポインタ17を移動し、そこでクリック操作をすることによりカーソル18を表示させる。

【0019】次に、例えばファイルアイコン13に対応するファイルを共有ウィンドウ12内に展開したい場合には、ファイルアイコン13上にポインタ17を移動し

てドラッグ操作を開始する。ドラッグ操作を開始すると、このドラッグ中は、移動するポインタ17がある位置にドラッグアイコン16が表示される。

【0020】そして、ポインタ17をカーソル18の位置に移動してドロップ操作（ドラッグ操作の後でマウスボタンを離すこと）をすると、上記ドラッグ操作により選択されたファイルアイコン13に対応するファイルの内容が、カーソル18の位置の前に挿入される。

【0021】また、ドロップ操作をすると、ドラッグ元のファイルアイコン13の表示およびこのファイルアイコン13に対応するファイルの内容は残るが、共有ウィンドウ12上のドラッグアイコン16の表示は消える。

【0022】テキストの展開位置を指定するには、図2(b)に示すように、ハッチング19を表示させることによっても行うことができる。この場合には、共有ウィンドウ12内でテキストを展開したい範囲にポインタ17を移動し、そこでドラッグ操作をすることによりハッチング19を表示させる。

【0023】そして、上述と同様にしてファイルアイコン13をドラッグしてこれをハッチング19の位置にドロップすると、ハッチング19の範囲内のテキストが、上記ドラッグ操作により選択されたファイルアイコン13に対応するファイルの内容で置換される。

【0024】図3は、上述のようなファイル展開後における表示部7の画面の表示例を示す図である。図2を用いて説明したように、ファイルアイコン13をカーソル18上でドロップすると、カーソル18の位置の前に上記ファイルアイコン13に対応するファイルの内容が挿入される。また、ファイルアイコン13をハッチング19上でドロップすると、ハッチング19の範囲内のテキストが上記ファイルアイコン13に対応するファイルの内容で置換される。

【0025】この結果、共有ウィンドウ12に展開されたテキストは、図3に示すようになる。なお、図3において、展開前の元のテキストは「XXX...XXX」で示され、展開によって挿入または置換されたファイルは「AAA...AAA」で示される。

【0026】上述のように操作部5により指定されたテキストを展開する位置または範囲（カーソル18の位置またはハッチング19の範囲）の情報、および指定されたファイルアイコン13に対応するファイル中のテキストは、回線制御部8により回線9を介して相手装置に送信される。

【0027】一方、上述のような送信側装置に回線9を介して接続されている受信側装置においても、表示画面11上には、相手側装置と同様の共有ウィンドウ12が表示されている。ただし、ファイルアイコン13～15は、必ずしも表示されているとは限らない。

【0028】これにより、受信側装置の回線制御部8によってテキストの展開位置または範囲の情報とファイル

中のテキストとが受信されると、送信側装置の共有ウィンドウ12内に展開された位置と同じ位置に、同じテキスト「AAA...AAA」が表示される。

【0029】図4は、本実施例におけるRAM3上のメモリマップを示す図である。図4において、Mdはドラッグの動作モード（「範囲指定」または「テキスト展開」）を格納する変数、Txは共有ウィンドウ12の左端からポインタ17までの距離を文字単位で格納する変数、Tyは共有ウィンドウ12の上端からポインタ17までの距離を文字単位で格納する変数である。

【0030】次いで、Sx(1)は共有ウィンドウ12の左端からカーソル18またはハッチング19の先頭までの距離を文字単位で格納する配列、Sx(2)は共有ウィンドウ12の左端からハッチング19の終端までの距離を文字単位で格納する配列（カーソル18が表示されているときは、ゼロに設定される）である。

【0031】次いで、Sy(1)は共有ウィンドウ12の上端からカーソル18、またはハッチング19の先頭までの距離を文字単位で格納する配列、Sy(2)は共有ウィンドウ12の上端からハッチング19の終端までの距離を文字単位で格納する配列（カーソル18が表示されているときは、ゼロに設定される）である。

【0032】次いで、Rx(1)は受信した配列Sx(1)の格納フィールドCx(1)の値を格納する変数、Rx(2)は受信した配列Sx(2)の格納フィールドCx(2)の値を格納する変数、Ry(1)は受信した配列Sy(1)の格納フィールドCy(1)の値を格納する変数、Ry(2)は受信した配列Sy(2)の格納フィールドCy(2)の値を格納する変数である。なお、上記各格納フィールドCx(1)、Cx(2)、Cy(1)、Cy(2)については後述する。

【0033】次いで、Pxは表示画面11の左端からポインタ17までの距離を画素単位で格納する変数、Pyは表示画面11の上端からポインタ17までの距離を画素単位で格納する変数、Dxは表示画面11の左端からドラッグアイコン16までの距離を画素単位で格納する変数、Dyは表示画面11の上端からドラッグアイコン16までの距離を画素単位で格納する変数である。

【0034】次いで、Fn(i)は展開するファイルのファイル名を格納する配列、Tx(i,j)は共有ウィンドウ12内に表示されるテキストを格納する配列（iは列番号、jは行番号）、Bf(i)はファイルアイコン13～15のビットマップを格納する配列、Bd(i)はドラッグアイコン16のビットマップを格納する配列である。

【0035】なお、本装置の起動時には、配列Bf(i)はファイルアイコン13～15のビットマップで初期化され、配列Bd(i)はドラッグアイコン16のビットマップで初期化されるものとする。そして、これらの初期化されたビットマップ情報は、各アイコン13～16表示のときに表示制御部6に渡される。

【0036】図5は、回線制御部8から送信されるテキストファイルのパケットのフォーマットを示す図である。図5において、Lpはパケットの長さを格納するフィールド、Cmはコマンドの種別を格納するフィールド、Cx(1)は配列Sx(1)の値を格納するフィールド、Cx(2)は配列Sx(2)の値を格納するフィールド、Cy(1)は配列Sy(1)の値を格納するフィールド、Cy(2)は配列Sy(2)の値を格納するフィールド、T(i)はファイルFn(i)内のテキストを格納するフィールドである。

【0037】次に、図6～図10に、図1に示したROM2に格納されているプログラムのうち、テキスト展開をする場合における各イベントの発生時にCPU1によって起動される処理のフローチャートを示す。

【0038】図6は、マウスのクリック時に起動される処理のフローチャートである。すなわち、このフローチャートは、図2(a)のカーソル18を表示させる際の処理を示している。

【0039】図6において、まずステップP1で、表示制御部6から読み込んだ現在選択中のオブジェクトの種別が「テキスト」か否かを判定し、「テキスト」であるならステップP2に進む。そうでないなら処理を終了する。

【0040】ステップP2では、CPU1は、配列Sx(1), Sx(2), Sy(1), Sy(2)が示す位置のカーソル18またはハッチング19の表示を消去するように表示制御部6に指示する。これは、カーソル18やハッチング19がすでに表示されているときに、それを消去するためである。

【0041】次に、ステップP3で、表示制御部6からポインタ17の位置情報を文字単位で変数TxとTyに読み込む。そして、ステップP4で、上記変数TxとTyの値をそれぞれ配列Sx(1)とSy(1)に設定し、ステップP5で、配列Sx(2)とSy(2)に値ゼロをそれぞれ設定する。続いて、ステップP6で、変数TxとTyが示す位置、すなわちクリック操作で指定した位置にカーソル18を表示するように表示制御部6に指示して、処理を終了する。

【0042】図7は、マウスのドラッグ操作の開始時に起動される処理のフローチャートである。すなわち、図7中のステップP1～P7の処理は、図2(b)のハッチング19の先頭位置を指定する際の処理を示し、ステップP8～P13の処理は、図2(a)および(b)に示したファイルアイコン13～15の中から所望のファイルアイコンを選択する際の処理を示している。

【0043】図7において、まずステップP1で、表示制御部6から読み込んだ現在選択中のオブジェクトの種別が「テキスト」か否かを判定し、「テキスト」であるならステップP2に進む。そうでないならステップP8に進む。

【0044】ステップP2では、変数Mdの値を「範囲指定」を表す値に設定する。続いてCPU1は、ステップP3で、配列Sx(1), Sx(2), Sy(1), Sy(2)が示す位置のカーソル18またはハッチング19の表示を消去するように表示制御部6に指示する。

【0045】次に、ステップP4で、表示制御部6からポインタ17の位置情報を文字単位で変数TxとTyに読み込む。そして、ステップP5で、変数TxとTyの値をそれぞれ配列Sx(1)とSy(1)に設定し、ステップP6で、変数TxとTyの値をそれぞれ配列Sx(2)とSy(2)に設定する。

【0046】その後、ステップP7で、配列Sx(1), Sx(2), Sy(1), Sy(2)が示す範囲に、すなわちドラッグ操作により指定した範囲にハッチング19を表示するように表示制御部6に指示し、処理を終了する。なお、この場合、Sx(1)=Sx(2), Sy(1)=Sy(2)であるから、ハッチング19はこの段階では点のように表示される。

【0047】一方、ステップP8では、表示制御部6から読み込んだ現在選択中のオブジェクトの種別が「ファイルアイコン」か否かを判定し、「ファイルアイコン」であるならステップP9に進む。そうでないならステップP14に進む。

【0048】ステップP9では、変数Mdの値を「テキスト展開」を表す値に設定する。続いて、ステップP10で、表示制御部6からポインタ17の位置情報を画素単位で変数PxとPyに読み込む。そして、ステップP11で、変数PxとPyの値をそれぞれ変数DxとDyに設定する。

【0049】次に、ステップP12で、表示制御部6から展開するファイルのファイル名を配列Fn(i)に読み込む。そして、ステップP13で、変数PxとPyが示す位置、すなわちポインタ17が存在している位置にドラッグアイコン16を表示するように表示制御部6に指示して、処理を終了する。

【0050】また、上記ステップP8からステップP14に進んだときは、「範囲指定」も「テキスト展開」も行わない状態を表す「アイドル」に対応する値に変数Mdの値を設定して、処理を終了する。

【0051】図8は、マウスをドラッグしながら移動している時に起動される処理のフローチャートである。すなわち、図8中のステップP1～P5の処理は、図2(b)のハッチング19の範囲を指定している際中の処理を示し、ステップP6～P10の処理は、図2(a)および(b)に示したドラッグアイコン16の移動中ににおける処理を示している。

【0052】図8において、まずステップP1で、変数Mdの値が「範囲指定」を表す値であるかどうかを判定し、「範囲指定」の値であるならステップP2に進む。

【0053】そうでないならステップP6に進む。

【0053】ステップP2では、配列Sx(1), Sx(2), Sy(1), Sy(2)の現在の値が示している範囲のハッチング19の表示を消去するように表示制御部6に指示する。続いて、ステップP3で、表示制御部6からポインタ17の位置情報を文字単位で変数TxとTyに読み込む。

【0054】次に、ステップP4で、変数TxとTyの値をそれぞれ配列Sx(2)とSy(2)に設定する。そして、ステップP5で、以上のようにして設定した配列Sx(1), Sx(2), Sy(1), Sy(2)の新たな値（ただし、ハッチング19の始点を表す配列Sx(1), Sy(1)の値は変わっていない）が示す範囲にハッチング19を表示するように表示制御部6に指示して、処理を終了する。

【0055】一方、ステップP1からステップP6に進んだときは、変数Mdの値が「テキスト展開」を表す値であるかどうかを判定し、「テキスト展開」の値であるならステップP7に進む。そうでないなら処理を終了する。

【0056】ステップP7では、変数DxとDyの現在の値が示している位置にあるドラッグアイコン16の表示を消去するように表示制御部6に指示する。続いて、ステップP8で、表示制御部6からポインタ17の位置情報を画素単位で変数PxとPyに読み込む。

【0057】次に、ステップP9で、変数PxとPyの値をそれぞれ変数DxとDyに設定する。そして、ステップP10で、以上のようにして設定した変数DxとDyの新たな値が示す位置にドラッグアイコン16を表示するように表示制御部6に指示して、処理を終了する。

【0058】図9は、マウスのドロップ時に起動される処理のフローチャートである。図9において、まずステップP1で、変数Mdの値が「テキスト展開」を表す値であるかどうかを判定し、「テキスト展開」の値であるならステップP2に進む。そうでないなら処理を終了する。

【0059】ステップP2では、変数DxとDyが示す位置にあるドラッグアイコン16の表示を消去するように表示制御部6に指示する。続いて、ステップP3で、変数DxとDyが示す位置が、共有ウィンドウ12内であるか否かを判定する。そして、共有ウィンドウ12内にあるならステップP4に進む。そうでないなら処理を終了する。

【0060】なお、ここでは、表示制御部6から共有ウィンドウ12の位置情報を画素単位で読み込み、この読み込んだ位置情報が以下の条件を満たすときは「共有ウィンドウ12内」であるとみなす。

共有ウィンドウの左端≤Dx≤共有ウィンドウの右端
共有ウィンドウの下端≤Dy≤共有ウィンドウの上端

【0061】次に、ステップP4で、表示制御部6からポインタ17の位置情報を文字単位で変数TxとTyに読み込む。そして、ステップP5で、変数TxとTyが

示す位置が、配列Sx(1), Sx(2), Sy(1), Sy(2)が示す位置か範囲内、すなわちカーソル18の位置またはハッチング19の範囲内であるかどうかを判定する。そして、これらの位置または範囲内にあるならステップP6に進む。そうでないなら処理を終了する。

【0062】なお、ここでは、表示制御部6から共有ウィンドウ12の位置情報を文字単位で読み込み、この読み込んだ位置情報が以下の条件を満たすときは「カーソル18の位置またはハッチング19の範囲内」であるとみなす。

【0063】Sx(2)=0, Sy(2)=0のとき

Tx=Sx(1)かつTy=Sy(1)

Sx(2)≠0, Sy(2)≠0であって

Ty=Sy(1)のとき、

Sx(1)≤Tx≤共有ウィンドウの右端

Sy(1)<Ty<Sy(2)のとき、

共有ウィンドウの左端≤Tx≤共有ウィンドウの右端

Ty=Sy(2)のとき、

共有ウィンドウの左端≤Tx≤Sx(2)

【0064】次に、ステップP6で、配列Sx(1), Sx(2), Sy(1), Sy(2)が示す位置または範囲にあるカーソル18またはハッチング19の表示を消去するように表示制御部6に指示する。続いて、ステップP7で、配列Sx(1), Sx(2), Sy(1), Sy(2)が示す位置または範囲にファイルFn(i)内のテキストを展開する。

【0065】ここで、Sx(2)=0, Sy(2)=0のときは、変数TxとTyが示す位置にファイルFn(i)内のテキストを挿入する。また、Sx(2)≠0, Sy(2)≠0のときは、配列Sx(1), Sx(2), Sy(1), Sy(2)

30 が示す範囲のテキストをファイルFn(i)内のテキストで置き換える。

【0066】そして、ステップP8で、テキスト展開コマンドをフィールドCmに格納し、配列Sx(1), Sx(2), Sy(1), Sy(2)の値をそれぞれフィールドCx(1), Cx(2), Cy(1), Cy(2)に格納し、ファイルFn(i)内のテキストをフィールドDt(i)に格納する。そして、これらの各フィールド情報を回線制御部8を介して相手装置に送信して、処理を終了する。

【0067】図10は、回線制御部8がフィールドCm中にテキスト展開コマンドを含むパケットを受信した時に起動される処理のフローチャートである。まずステップP1で、回線制御部8から受信したフィールドCx(1), Cx(2), Cy(1), Cy(2)の情報をそれぞれ配列Rx(1), Rx(2), Ry(1), Ry(2)に設定する。次に、ステップP2で、回線制御部8から受信したフィールドDt(i)の情報を配列Ry(1), Ry(2), Rx(1), Rx(2)が示す位置または範囲に展開して、処理を終了する。

【0068】このように、本実施例では、同じ表示画面11上に表示されたファイルアイコン13に対してドラッグ操作およびドロップ操作をするだけで、上記ファイ

ルアイコン13に対応するファイル内のテキストを共有ウィンドウ12に展開するとともに、相手装置に送信することができ、共有ウィンドウ12にテキストを書き込む操作を簡便に行うことができる。

【0069】なお、前述した第1の実施例では、テキストの指定範囲をハッチング19で表示するようにしていて、アンダーライン、太字または斜体等で表示するようにしてよい。

【0070】次に、第2の実施例として、ファイル内の画像を共有ウィンドウに展開する場合について説明する。なお、本実施例の通信装置の構成は、図1に示したものと同じである。

【0071】図11は、画像のファイルを展開する前ににおける表示部7の画面の表示例を示す図であり、図2に示したものと同じものには同じ符号を付している。図11において、21～23はファイルアイコンであり、図1の外部記憶装置4に格納されている種々の画像のファイルに対応するものである。

【0072】次いで、24はクリック操作により表示されたカーソルであり、上記テキスト展開時に表示されるカーソル18(図2(a)参照)とは異なる形状をしている。また、25はドラッグ操作により共有ウィンドウ12内に表示されたフレームである。

【0073】図11(a)に示すように、外部記憶装置4に格納されているファイル中の画像を共有ウィンドウ12内に表示したいときは、まず、共有ウィンドウ12内で画像を表示したい位置にポインタ17を移動し、そこでクリック操作することによりカーソル24を表示させる。

【0074】次に、例えばファイルアイコン21に対応するファイル中の画像を共有ウィンドウ12内に表示したい場合には、ファイルアイコン21上にポインタ17を移動してドラッグ操作を開始する。ドラッグ操作を開始すると、このドラッグ中は、移動するポインタ17がある位置にドラッグアイコン16が表示される。

【0075】その後、ポインタ17をカーソル24の位置に移動してドロップ操作をすると、上記ドラッグ操作により選択されたファイルアイコン21に対応するファイル中の画像が、カーソル24の位置に表示される。

【0076】また、ドロップ操作をすると、ドラッグ元のファイルアイコン21の表示およびこのファイルアイコン21に対応するファイルの内容は残るが、共有ウィンドウ12上のドラッグアイコン16の表示は消える。

【0077】画像の表示位置を指定するには、図11(b)に示すように、フレーム25を表示させることによっても行うことができる。すなわち、共有ウィンドウ12内の画像を表示したい範囲にポインタ17を移動し、そこでドラッグ操作することによりフレーム25を表示させる。この場合のドラッグ操作は、フレーム25の対角線方向に行われる。

【0078】そして、上述と同様にしてファイルアイコン21をドラッグしてこれをフレーム25の位置にドロップすると、上記ドラッグ操作により選択されたファイルアイコン21に対応する画像がフレーム25の範囲内に表示される。

【0079】図12は、上述のようなファイル展開後における表示部7の画面の表示例を示す図である。図12に示すように、ファイルアイコン21をカーソル24上またはフレーム25上でドロップすると、カーソル18の位置またはフレーム25の範囲内に上記ファイルアイコン21に対応する画像26が表示される。

【0080】上述のように操作部5により指定された画像を表示する位置または範囲(カーソル24の位置またはフレーム25の範囲)の情報、および指定されたファイルアイコン13に対応するファイル中の画像データは、回線制御部8により回線9を介して相手装置に送信される。

【0081】一方、上述のような送信側装置に回線9を介して接続されている受信側装置においても、表示画面20 11上には、相手側装置と同様の共有ウィンドウ12が表示されている。ただし、ファイルアイコン21～23は、必ずしも表示されているとは限らない。

【0082】これにより、受信側装置の回線制御部8によって画像の表示位置または範囲の情報とファイル中の画像データとが受信されると、送信側装置の共有ウィンドウ12内に表示された位置または範囲と同じ位置または範囲に、同じ画像26が表示される。

【0083】図13は、本実施例におけるRAM3上のメモリマップを示す図である。なお、図13において、

30 図4に示したメモリマップ中に示したものと同じ符号を付したものは、同様の内容を有する変数または配列である。まず、Md'はドラッグの動作モード(「範囲指定」または「画像展開」)を格納する変数である。

【0084】次いで、Wx(1)は表示画面11の左端から共有ウィンドウ12の左端までの距離を画素単位で格納する配列、Wx(2)は表示画面11の左端から共有ウィンドウ12の右端までの距離を画素単位で格納する配列、Wy(1)は表示画面11の上端から共有ウィンドウ12の上端までの距離を画素単位で格納する配列、Wy(2)は表示画面11の上端から共有ウィンドウ12の下端までの距離を画素単位で格納する配列である。

【0085】次いで、Sx(1)'は共有ウィンドウ12の左端からカーソル24またはフレーム25の左端までの距離を画素単位で格納する配列、Sx(2)'は共有ウィンドウ12の左端からフレーム25の右端までの距離を画素単位で格納する配列(カーソル24が表示されているときは、ゼロに設定される)である。

【0086】次いで、Sy(1)'は共有ウィンドウ12の上端からカーソル24またはフレーム25の左端までの距離を画素単位で格納する配列、Sy(2)'は共有ウィン

ドウ12の上端からフレーム25の右端までの距離を画素単位で格納する配列（カーソル24が表示されているときは、ゼロに設定される）である。

【0087】次いで、R_x(1)'は受信した配列S_x(1)'の格納フィールドF_x(1)の値を格納する変数、R_x(2)'は受信した配列S_x(2)'の格納フィールドF_x(2)の値を格納する変数、R_y(1)'は受信した配列S_y(1)'の格納フィールドF_y(1)の値を格納する変数、R_y(2)'は受信した配列S_y(2)'の格納フィールドF_y(2)の値を格納する変数である。なお、上記各格納フィールドF_x(1)、F_x(2)、F_y(1)、F_y(2)については後述する。

【0088】次いで、B_xは共有ウィンドウ12内に表示されるファイル中の画像26の横方向に対する画素数を格納する変数、B_yは共有ウィンドウ12内に表示されるファイル中の画像26の縦方向に対する画素数を格納する変数、B(i, j)は共有ウィンドウ12内に表示されるファイル中の画像26のビットマップを格納する配列（iは横方向の画素の番号、jは縦方向の画素の番号）である。

【0089】次いで、B_f_xはファイルアイコン21～23の画像の横方向に対する画素数を格納する変数、B_f_yはファイルアイコン21～23の画像の縦方向に対する画素数を格納する変数、B_f(i, j)はファイルアイコン21～23の画像のビットマップを格納する配列（iは横方向の画素の番号、jは縦方向の画素の番号）である。

【0090】次いで、B_d_xはドラッグアイコン16の画像の横方向に対する画素数を格納する変数、B_d_yはドラッグアイコン16の画像の縦方向に対する画素数を格納する変数、B_d(i, j)はドラッグアイコン16の画像のビットマップを格納する配列（iは横方向の画素の番号、jは縦方向の画素の番号）である。

【0091】なお、本装置の起動時には、変数B_f_xはファイルアイコン21～23の画像の横方向に対する画素数で初期化され、変数B_f_yはファイルアイコン21～23の画像の縦方向に対する画素数で初期化され、配列B_f(i, j)はファイルアイコン21～23の画像のビットマップで初期化されるものとする。

【0092】また、変数B_d_xはドラッグアイコン16の画像の横方向に対する画素数で初期化され、変数B_d_yはドラッグアイコン16の画像の縦方向に対する画素数で初期化され、配列B_d(i, j)はドラッグアイコン16の画像のビットマップで初期化されるものとする。

【0093】そして、これらの初期化された変数および配列は、各アイコンを表示するときに表示制御部6に渡される。また、共有ウィンドウ12が表示されるとき、配列W_xとW_yは、表示画面11内の共有ウィンドウ12の位置で初期化されるものとする。

【0094】図14は、回線制御部8から送信される画

像ファイルのパケットのフォーマットを示す図である。図14において、L_pはパケットの長さを格納するフィールド、C_mはコマンドの種別を格納するフィールド、F_x(1)は配列S_x(1)'の値を格納するフィールド、F_x(2)は配列S_x(2)'の値を格納するフィールド、F_y(1)は配列S_y(1)'の値を格納するフィールド、F_y(2)は配列S_y(2)'の値を格納するフィールド、B_m(i, j)は表示されたファイル中の画像26のビットマップを格納するフィールド（iは横方向の画素の番号、jは縦方向の画素の番号）である。

【0095】次に、図15～図19に、ROM2に格納されているプログラムのうち、画像展開をする場合における各イベントの発生時にCPU1によって起動される処理のフローチャートを示す。

【0096】図15は、マウスのクリック時に起動される処理のフローチャートである。すなわち、このフローチャートは、図11(a)のカーソル24を表示させる際の処理を示している。

【0097】図15において、まずステップP1で、表示制御部6から読み込んだ現在選択中のオブジェクトの種別が「共有ウィンドウ」か否かを判定し、「共有ウィンドウ」であるならステップP2に進む。そうでないなら処理を終了する。

【0098】次に、ステップP2～P6の処理は、図6に示したステップP2～P6の処理とほぼ同じである。異なるところは、図6のフローチャートでは、テキストを扱っていたために文字単位の変数T_x、T_yと文字単位の配列S_x(1), S_x(2), S_y(1), S_y(2)とを用いていたが、本実施例では、画像を扱っているために画素単位の変数P_x、P_yと画素単位の配列S_x(1)', S_x(2)', S_y(1)', S_y(2)'とを用いていることのみである。よって、ここでは処理内容の詳細な説明は省略することとする。

【0099】図16は、マウスのドラッグ操作の開始時に起動される処理のフローチャートである。すなわち、図16中のステップP1～P7の処理は、図11(b)に示したフレーム25の先頭位置を指定する際の処理を示し、ステップP8～P13の処理は、図11(a)および(b)に示したファイルアイコン21～23の中から所望のファイルアイコンを選択する際の処理を示している。

【0100】図16において、まずステップP1で、表示制御部6から読み込んだ現在選択中のオブジェクトの種別が「共有ウィンドウ」か否かを判定し、「共有ウィンドウ」であるならステップP2に進む。そうでないならステップP8に進む。

【0101】この図16も上記した図15と同様に、ステップP2～P14の処理は、図7に示したステップP2～P14の処理とほぼ同じである。すなわち、本実施例では、ドラッグの動作モードを格納する変数としてM

d' を用いるとともに、画素単位の変数 P_x , P_y と画素単位の配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ とを用いていることのみが、図 7 のフローチャートと相違する。よって、ここでも処理内容の詳細な説明は省略することとする。

【0102】図 17 は、マウスをドラッグしながら移動している時に起動される処理のフローチャートである。

すなわち、図 17 中のステップ P 1～P 6 の処理は、図 11 (b) に示したフレーム 25 の範囲を指定している際中の処理を示し、ステップ P 7～P 11 の処理は、図 11 (a) および (b) に示したドラッグアイコン 16 の移動中における処理を示している。

【0103】図 17において、まずステップ P 1 で、変数 Md' の値が「範囲指定」を表す値であるかどうかを判定し、「範囲指定」の値であるならステップ P 2 に進む。そうでないならステップ P 7 に進む。

【0104】ステップ P 2 では、表示制御部 6 からポインタ 17 の位置情報を画素単位で変数 P_x と P_y に読み込む。そして、ステップ P 3 で、この読み込んだ変数 P_x と P_y が示す位置が、配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ が示す位置または範囲内にあるかどうかを判定し、範囲内にあるならステップ P 4 に進む。そうでないなら処理を終了する。

【0105】なお、ここでは、以下の条件を満たすときに、「配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ が示す位置または範囲内」であるとみなす。

【0106】 $S_x(1)' \leq S_x(2)'$ 、かつ、 $S_y(1)' \leq S_y(2)'$ のとき、 $W_x(1) + S_x(1)' \leq P_x \leq W_x(1) + S_x(2)'$ 、かつ、 $W_y(1) + S_y(1)' \leq P_y \leq W_y(1) + S_y(2)'$

$S_x(1)' > S_x(2)'$ 、かつ、 $S_y(1)' < S_y(2)'$ のとき、 $W_x(1) + S_x(2)' \leq P_x \leq W_x(1) + S_x(1)'$ 、かつ、 $W_y(1) + S_y(1)' \leq P_y \leq W_y(1) + S_y(2)'$

【0107】 $S_x(1)' < S_x(2)'$ 、かつ、 $S_y(1)' > S_y(2)'$ のとき、 $W_x(1) + S_x(1)' \leq P_x \leq W_x(1) + S_x(2)'$ 、かつ、 $W_y(1) + S_y(2)' \leq P_y \leq W_y(1) + S_y(1)'$

$S_x(1)' > S_x(2)'$ 、かつ、 $S_y(1)' > S_y(2)'$ のとき、 $W_x(1) + S_x(2)' \leq P_x \leq W_x(1) + S_x(1)'$ 、かつ、 $W_y(1) + S_y(2)' \leq P_y \leq W_y(1) + S_y(1)'$

【0108】次に、ステップ P 4 で、配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ の現在の値が示している範囲のフレーム 25 の表示を消去するように表示制御部 6 に指示する。続いて、ステップ P 5 で、変数 P_x と P_y の値をそれぞれ配列 $S_x(2)', S_y(2)'$ に設定する。

【0109】そして、ステップ P 5 で、以上のようにして設定した配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ の新たな値が示す範囲にフレーム 25 を表示するように表示制御部 6 に指示して、処理を終了する。ここで、配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ が作

る線分が対角線となるようにフレーム 25 を表示する。

【0110】一方、ステップ P 1 からステップ P 7 に進んだときは、変数 Md' の値が「画像展開」を表す値であるかどうかを判定し、「画像展開」の値であるならステップ P 8 に進む。そうでないなら処理を終了する。なお、ステップ P 8～P 11 の処理は、図 8 に示したフローチャートのステップ P 7～P 10 の処理の同様であるので、説明を省略する。

【0111】図 18 は、マウスのドロップ時に起動される処理のフローチャートである。図 18において、まずステップ P 1 で、変数 Md' の値が「画像展開」を表す値であるかどうかを判定し、「画像展開」の値であるならステップ P 2 に進む。そうでないなら処理を終了する。

【0112】ステップ P 2 では、変数 D_x と D_y が示す位置のドラッグアイコン 16 の表示を消去するように表示制御部 6 に指示する。続いて、ステップ P 3 で、表示制御部 6 からポインタ 17 の位置情報を画素単位で変数 P_x と P_y に読み込む。

【0113】そして、ステップ P 4 で、この取り込んだ変数 P_x と P_y が示すポインタ 17 の位置が、配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ が示す位置または範囲内、すなわちカーソル 24 の位置またはフレーム 25 の範囲内であるかどうかを判定する。そして、これらの位置または範囲内にあるならステップ P 5 に進む。そうでないなら処理を終了する。

【0114】なお、ここでは、以下の条件を満たすときに、「カーソル 24 の位置またはフレーム 25 の範囲内」であるとみなす。

30 $S_x(2)' = 0, S_y(2)' = 0$ のとき
 $P_x = W_x(1) + S_x(1)'$ 、かつ、 $P_y = W_y(1) + S_y(1)'$
 $S_x(2)' \neq 0, S_y(2)' \neq 0$ であって、図 17 のステップ P 3 と同じ条件を満たすとき

【0115】次に、ステップ P 5 で、配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ が示す範囲のフレーム 25 の表示を消去するように表示制御部 6 に指示する。続いて、ステップ P 6 で、ファイル名 $F_n(i)$ に対応する画像 26 の横方向および縦方向の画素数をそれぞれ変数 B_x と B_y に読み込む。

【0116】そして、ステップ P 7 で、配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ が示す位置または範囲にファイル名 $F_n(i)$ に対応する画像 26 を表示する。ここで、 $S_x(2)' = 0, S_y(2)' = 0$ のときは、変数 $S_x(2)'$ に $(W_x(1) + B_x)$ と $W_x(2)$ の小さい方の値を設定するとともに、変数 $S_y(2)'$ に $(W_y(1) + B_y)$ と $W_y(2)$ の小さい方の値を設定する。これは、配列 $S_x(1)', S_x(2)', S_y(1)', S_y(2)'$ により表示される画像を表示する範囲 26 が、共有ウィンドウ 12 からはみ出さないようにするためである。

【0117】また、ファイルF n(i)から読み込んだ配列B(i,j)で示される画像26は、横方向に|Sx(1)' - Sx(2)'|/Bx倍されるとともに、縦方向に|Sy(1)' - Sy(2)'|/By倍される。ここで、|α|はα式の絶対値を表す。

【0118】次に、ステップP8で、画像展開コマンドをフィールドCmに格納し、配列Sx(1)', Sx(2)', Sy(1)', Sy(2)'の値をそれぞれフィールドFx(1), Fx(2), Fy(1), Fy(2)に格納し、配列B(i,j)で示される画像の値をフィールドBb(i,j)に格納する。そして、これらの各フィールド情報を回線制御部8を介して相手装置に送信して、処理を終了する。

【0119】図19は、回線制御部8がフィールドCm中に画像展開コマンドを含むパケットを受信した時に起動される処理のフローチャートである。まずステップP1で、回線制御部8から受信したフィールドCx(1), Cx(2), Cy(1), Cy(2)の情報をそれぞれ配列Rx(1)', Rx(2)', Ry(1)', Ry(2)'に格納する。次に、ステップP2で、回線制御部8から受信したフィールドT(i)の情報を、配列Rx(1)', Rx(2)', Ry(1)', Ry(2)'が示す位置または範囲に展開して、処理を終了する。

【0120】このように、本実施例では、同じ表示画面11上に表示されたファイルアイコン21に対してドラッグ操作およびドロップ操作をするだけで、上記ファイルアイコン21に対応するファイル内の画像を共有ウィンドウ12に表示するとともに、これを相手装置に送信することができ、共有ウィンドウ12に画像を表示する操作を簡便に行うことができる。

【0121】なお、前述した第2の実施例では、画像の表示範囲をフレーム25で表示するようにしていたが、ハッチングを付けたり表示の色を変えることによって表示するようにしてもよい。

【0122】また、前述した第2の実施例では、クリックした位置が画像26の左上端になるようにしているが、クリックした位置が画像26の中央や右下端などになるようにしてもよい。

【0123】また、図18のステップP7において、配列B(i,j)で示される画像26をフレーム25に合わせて拡大縮小するときの縦横の比率が異なるときがあるが、以下のようにすることによって対応することができる。このようにすれば、縦横の比率が保たれる。

【0124】 $(|Sx(1)' - Sx(2)'| / Bx) \leq (|Sy(1)' - Sy(2)'| / By)$ のとき、縦横方向にそれぞれ|Sx(1)' - Sx(2)'|/Bx倍する。

$(|Sx(1)' - Sx(2)'| / Bx) > (|Sy(1)' - Sy(2)'| / By)$ のとき、縦横方向にそれぞれ|Sy(1)' - Sy(2)'|/By倍する。

【0125】以上、第1の実施例および第2の実施例で述べたように、回線9を介して接続された送信側装置と

受信側装置との間で、共有ウィンドウ12内に展開または表示されるテキストや画像などのファイルを相互に転送することにより、これらのファイルを共有することができるようになされている。

【0126】ところが、従来の通信装置では、通信装置の使用中にファイルを相手装置に転送するときは、ファイル転送アプリケーションを別に起動し、相手先のファイル名等を指定して転送を行うか、または送信したいファイルのアイコンを共有ウィンドウ内にドラッグおよびドロップして転送を行っていた。

【0127】しかしながら、このような従来例では、通信装置の使用中にファイルを転送するときは、受信側装置でファイル受信が完了したかどうかを送信側装置において確認することができないという欠点があった。次に述べる第3の実施例による通信装置は、このような欠点を解決するためになされたものである。

【0128】図20は、ファイル送信時における表示部7の画面の表示例を示す図である。図20において、31～33はファイルアイコンであり、送信側装置の外部記憶装置4に保存されているテキストや画像等のファイルに対応するものである。また、34はドロップ操作後共有ウィンドウ12内に表示されるファイルアイコンである。

【0129】ファイルを相手装置に送信したいときは、送信したいファイルに対応するファイルアイコン31をドラッグし、その後、共有ウィンドウ12上でドロップすることによって行う。

【0130】すなわち、まず、送信したいファイルに対応するファイルアイコン31上にポインタ17を移動してドラッグ操作を開始する。このドラッグ中は、移動するポインタ17の位置にドラッグアイコン16が表示される。

【0131】その後、ポインタ17を共有ウィンドウ12内に移動してドロップ操作をすると、ドロップ操作した位置にファイルアイコン34が表示されるとともに、ドラッグアイコン16の表示は消える。そして、ファイルアイコン34に対応するファイルが相手装置に送信される。

【0132】このとき、ドラッグ元のファイルアイコン31の表示されたままであり、これに対応するファイルの内容は保持される。その後、相手装置においてファイルの受信が完了すると、ファイルアイコン34の表示が消える。これにより、相手装置がファイルの受信を完了したかどうかを送信側で確認することができるようになる。

【0133】図21は、ファイル受信時における表示部7の画面の表示例を示す図である。図21において、35は送信側装置からファイルを受信したときに共有ウィンドウ12内に表示されるファイルアイコンである。また、36はドロップ操作後に表示されるファイルアイコ

ンである。

【0134】ファイルの受信があると、共有ウィンドウ12上にファイルアイコン35が表示される。このとき、受信したファイルは一時ファイルに記憶される。この一時ファイルに記憶されたファイルを保存したいときは、ファイルアイコン35をドラッグし、共有ウィンドウ12の外でドロップすることによって行う。

【0135】すなわち、ファイルアイコン35上にポイントタ17を移動してドラッグ操作を開始すると、ポイントタ17の位置にドラッグアイコン16が表示される。その後、ポイントタ17を共有ウィンドウ12の外に移動してドロップ操作をすると、ドロップ操作をした位置にファイルアイコン36が表示されるとともに、ファイルアイコン35およびドラッグアイコン16の表示は消える。これでファイルの受信が完了する。

【0136】図22は、本実施例におけるRAM3上のメモリマップを示す図である。なお、図22において、図4に示したメモリマップ中に示したものと同じ符号を付したもののは、同様の内容を有する変数または配列である。まず、Md''はファイル転送の動作モード（「送信」または「受信」）を格納する変数である。

【0137】次いで、Csはファイル送信要求コマンドを格納する変数、Crはファイル受信確認コマンドを格納する変数、Sxは変数Dxの値を格納する変数、Syは変数Dyの値を格納する変数である。

【0138】次いで、Rxは受信したフィールドSxの値を格納する変数、Ryは受信したフィールドSyの値を格納する変数、Fsは送信するファイルのファイル名を格納する配列、Frは受信したファイル名（フィールドFsの値）を格納する配列である。また、Bfはファイルアイコン31～36のビットマップを格納する配列、Bdはドラッグアイコン16のビットマップを格納する配列である。

【0139】なお、本装置の起動時には、変数Csはファイル送信要求コマンドで初期化され、変数Crはファイル受信確認コマンドで初期化されるものとする。また、配列Bfはファイルアイコン31～36のビットマップで初期化され、配列Bdはドラッグアイコン16のビットマップで初期化されるものとする。そして、これらの初期化された配列Bf、Bd中のビットマップ情報は、各アイコン表示時に表示制御部6に渡される。

【0140】図23は、送信側装置の回線制御部8から送信されるパケットのフォーマットを示す図である。図23において、Lpはパケットの長さを格納するフィールド、Csはファイル送信要求コマンド（変数Cs）を格納するフィールド、Sxは変数Sxを格納するフィールド、Syは変数Syを格納するフィールド、Fsは送信するファイルのファイル名（配列Fs）を格納するフィールド、Fcは送信するファイルの内容を格納するフィールドである。

【0141】また、図24は、受信側装置の回線制御部8から送信されるパケットのフォーマットを示す図である。図24において、Lpはパケットの長さを格納するフィールド、Crはファイル受信確認コマンド（変数Cr）を格納するフィールドである。

【0142】次に、図25～図29に、ROM2に格納されているプログラムのうち、本実施例における各イベントの発生時にCPU1によって起動される処理のフローチャートを示す。

10 【0143】図25は、マウスのドラッグ操作の開始時に起動される処理のフローチャートである。すなわち、このフローチャートは、図20のファイルアイコン31または図21のファイルアイコン35を指定する際の処理を示している。

【0144】図25において、まずステップP1で、表示制御部6から読み込んだ現在選択中のオブジェクトの種別が「ファイルアイコン」か否かを判定し、「ファイルアイコン」であるならステップP2に進む。そうでないなら処理を終了する。

20 【0145】ステップP2では、表示制御部6からポイントタ17の位置情報を変数PxとPyに読み込む。そして、ステップP3で、この読み込んだ変数PxとPyが示す位置にドラッグアイコン16を表示するように表示制御部6に指示する。

【0146】次に、ステップP4で、変数PxとPyの値をそれぞれ変数DxとDyに格納し、ステップP5で、上記変数PxとPyが示すポイントタ17の位置が共有ウィンドウ12内にあるかどうかを判定する。ここで、ポイントタ17の位置が共有ウィンドウ12内にない

30 ときは、図20のファイルアイコン31を指定した場合に相当するので、ステップP6に進む。一方、ポイントタ17の位置が共有ウィンドウ12内にあるときは、図21のファイルアイコン35を指定した場合に相当するので、ステップP8に進む。

【0147】なお、ここでは、表示制御部6から共有ウィンドウ12の位置情報を読み込み、この読み込んだ位置情報が以下の条件を満たすときに「共有ウィンドウ内」であるとみなす。

共有ウィンドウの左端 < Px < 共有ウィンドウの右端

40 共有ウィンドウの下端 < Py < 共有ウィンドウの上端

【0148】そして、上記ステップP5からステップP6に進んだときは、変数Md''の値を「送信」を表す値に設定する。そして、ステップP7で、表示制御部6からファイルアイコン31に対応するファイル名（ここでは「File1」）を配列Fsに読み込んで、処理を終了する。一方、上記ステップP5からステップP8に進んだときは、変数Md''の値を「受信」を表す値に設定して、処理を終了する。

【0149】図26は、マウスをドラッグしながら移動している時に起動される処理のフローチャートである。

図26において、まずステップP1で、変数DxとDyが示す位置にあるドラッグアイコン16の表示を消去するように表示制御部6に指示する。

【0150】次に、ステップP2で、表示制御部6から現在のポインタ17の位置情報を変数PxとPyに読み込む。そして、ステップP3で、この読み込んだ変数PxとPyが示す位置にドラッグアイコン16を表示するように表示制御部6に指示する。続いて、ステップP4で、変数PxとPyの値をそれぞれ変数DxとDyに格納して、処理を終了する。

【0151】図27は、マウスのドロップ時に起動される処理のフローチャートである。すなわち、図27中のステップP1～P6の処理は、図20のファイルアイコン31を共有ウィンドウ12内にドロップしてファイルアイコン34を表示する場合の処理を示している。また、ステップP7～P11の処理は、図21のファイルアイコン35を共有ウィンドウ12の外にドロップしてファイルアイコン36を表示する場合の処理を示している。

【0152】図27において、まずステップP1で、変数DxとDyが示す位置のドラッグアイコン16の表示を消去するように表示制御部6に指示する。次に、ステップP2で、変数Md''の値が「送信」を表す値であるかどうかを判定し、「送信」の値であるならステップP3に進む。そうでないならステップP7に進む。

【0153】ステップP3では、変数DxとDyが示す位置が共有ウィンドウ12内にあるかどうかを判定し、共有ウィンドウ12内にあるならステップP4に進む。そうでないなら処理を終了する。ここで、共有ウィンドウ12内にあるかどうかの判定は、図25のステップP5における判定と同じ条件で行われる。

【0154】上記ステップP3からステップP4に進んだときは、変数DxとDyが示す位置に、配列Fs中に読み込んだファイル名に対応するファイルアイコン34を表示するように表示制御部6に指示する。また、ステップP5で、変数DxとDyの値をそれぞれ変数SxとSyに格納する。

【0155】そして、ステップP6で、変数Cs、変数Sx、変数Sy、配列Fsの値および配列Fs中に読み込んだファイル名に対応するファイルの内容を、それぞれフィールドCs、Sx、Sy、Fs、Fcに設定し、これを回線制御部8に送信して処理を終了する。

【0156】一方、上記ステップP2からステップP7に進んだときは、変数DxとDyが示す位置が共有ウィンドウ12内にあるかどうかを判定し、共有ウィンドウ12内にあるならステップP8に進む。そうでないなら処理を終了する。このステップP7における共有ウィンドウ12内にあるかどうかの判定も、図25のステップP5における判定と同じ条件で行われる。

【0157】ステップP8では、変数RxとRyが示す

位置にあるファイルアイコン35の表示を消去するように表示制御部6に指示し、ステップP9では、変数DxとDyが示す位置に、配列Fs中に読み込んだファイル名に対応するファイルアイコン36を表示するように表示制御部6に指示する。

【0158】次に、ステップP10で、一時ファイルの名前を配列Frの値で示される名前に変更する。そして、ステップP11で、変数Crの値をフィールドCrに設定し、これを回線制御部8に送信して処理を終了する。

【0159】図28は、回線制御部8が図23のようなフィールドCsを含むパケットを受信した時に起動される処理のフローチャートである。まずステップP1で、回線制御部8から受信したフィールドSx、Sy、Fsの値をそれぞれ変数Rx、変数Ryおよび配列Frに格納する。

【0160】次に、ステップP2で、回線制御部8から受信したフィールドFcの値を一時ファイルに格納するとともに、ステップP3で、変数RxとRyが示す位置に、配列Fs中に読み込んだファイル名に対応するファイルアイコン35を表示するように表示制御部6に指示して、処理を終了する。

【0161】図29は、回線制御部8が図24のようなフィールドCrを含むパケットを受信した時に起動される処理のフローチャートである。この場合は、ステップP1で、変数SxとSyが示す位置のファイルアイコン34の表示を消去するように表示制御部6に指示して、処理を終了する。

【0162】なお、前述した第3の実施例では、送信側と受信側の共有ウィンドウ12は、画面11中の同じ位置にある場合について説明している。これに対し、送信側の共有ウィンドウ12内のファイルアイコン34の相対的な位置情報を受信側に送信するようにしてもよい。このようにすれば、共有ウィンドウ12の位置は、送信側と受信側とで必ずしも同じ位置にある必要はなくなる。

【0163】また、前述した第3の実施例では、図20のファイルアイコン34と図21のファイルアイコン35は、送信側と受信側とで共有ウィンドウ12中の同じ位置に表示されているが、ファイルアイコン35は、受信側の共有ウィンドウ12の固定された位置（例えば、左上端や中央など）に表示するようにしてもよい。

【0164】このとき、変数RxとRyは、ファイルアイコン35を表示する固定した位置を示す値で予め初期化される。また、フィールドSxとSyは必要でない。したがって、図27のステップP6においては、フィールドSxとSyの値は受信側装置に送信しない。また、図28のステップP1においては、フィールドSxとSyの値は受信せず、それぞれの値を変数RxとRyに格納することもない。

【0165】また、前述した第3の実施例では、受信したファイルの名前は送信側と同じになるが、ファイルアイコン35を共有ウィンドウ12の外にドラッグする前に、受信ファイルの新しい名前を受け付けるようにすることも可能である。

【0166】さらに、前述した第1～第3の実施例では、端末に接続される回線9としてLANを使用しているが、ISDNの基本インタフェース、ISDNの一次群インタフェース、広帯域ISDN、回線交換網、パケット交換網、公衆電話網、専用線、構内交換機(PBX)などでも実施することができる。

【0167】また、前述した第1～第3の実施例では、プログラムやデータを格納する記憶装置としてROM2やRAM3を使用しているが、フロッピーディスク、ハードディスク、ICカードなどでも実施することができる。

【0168】また、前述した第1～第3の実施例では、ファイルアイコン13～15をドラッグしてテキストの展開や画像の表示またはファイルの送信などを行うようになっていたが、フォルダアイコンをドラッグしてフォルダ内のテキストの展開したり、画像を表示したり、ファイルを送信したりするようにしてもよい。なお、フォルダとは複数のファイルをまとめて一つの名前を付けたものである。

【0169】また、前述した第1～第3の実施例では、テキストや画像は一つのパケットで送信していたが、テキストや画像のデータ量が大きいときに複数のパケットに分割して送信するようにしてもよい。

【0170】

【発明の効果】本発明は上述したように、種々のファイルに対応して表示されているアイコンに対して所定の操作、例えばドラッグ操作およびドロップ操作をすることにより、上記ファイル内の情報(例えばテキストや画像)を共有画面に展開して、上記ファイル内の情報を相手側装置と共有する手段を設けたので、エディタを一々起動せずに、上記アイコンに対してドラッグ操作およびドロップ操作をするだけで、上記アイコンに対応する情報を共有画面に取り込むことができ、ファイル内の情報を共有画面に取り込む操作を簡便に、かつ迅速に行うことができる。

【0171】本発明の他の特徴によれば、送信側装置から受信側装置にファイルを送信したときに、その送信ファイルに対応するファイルアイコンを送信側装置で表示し、受信側装置でファイル受信が完了したときに、上記ファイルアイコンの表示を消す手段を設けたので、通信装置の使用中にファイルを相手側装置に送信するときに、受信側装置でファイル受信を完了したかどうかを送信側において確認することができ、円滑な情報交換を行うことができるようになる。

【図面の簡単な説明】

【図1】本発明の通信装置の一実施例を示すブロック図である。

【図2】テキストファイル展開前における表示画面の表示例を示す図である。

【図3】テキストファイル展開後における表示画面の表示例を示す図である。

【図4】RAM上のメモリマップを示す図である。

【図5】回線制御部から送信されるパケットのフォーマットを示す図である。

10 【図6】マウスのクリック時に起動される処理のフローチャートである。

【図7】マウスのドラッグ操作の開始時に起動される処理のフローチャートである。

【図8】マウスをドラッグしながら移動している時に起動される処理のフローチャートである。

【図9】マウスのドロップ操作時に起動される処理のフローチャートである。

【図10】テキスト展開コマンドを含むパケットを受信した時に起動される処理のフローチャートである。

20 【図11】画像ファイル展開前における表示画面の表示例を示す図である。

【図12】画像ファイル展開後における表示画面の表示例を示す図である。

【図13】RAM上のメモリマップを示す図である。

【図14】回線制御部から送信されるパケットのフォーマットを示す図である。

【図15】マウスのクリック時に起動される処理のフローチャートである。

【図16】マウスのドラッグ操作の開始時に起動される処理のフローチャートである。

30 【図17】マウスをドラッグしながら移動している時に起動される処理のフローチャートである。

【図18】マウスのドロップ操作時に起動される処理のフローチャートである。

【図19】画像展開コマンドを含むパケットを受信した時に起動される処理のフローチャートである。

【図20】ファイル送信時における表示画面の表示例を示す図である。

40 【図21】ファイル受信時における表示画面の表示例を示す図である。

【図22】RAM上のメモリマップを示す図である。

【図23】回線制御部から送信されるパケットのフォーマットを示す図である。

【図24】回線制御部から送信されるパケットの他のフォーマットを示す図である。

【図25】マウスのドラッグ操作の開始時に起動される処理のフローチャートである。

【図26】マウスをドラッグしながら移動している時に起動される処理のフローチャートである。

50 【図27】マウスのドロップ操作時に起動される処理の

フローチャートである。

【図28】図23のパケットを受信した時に起動される処理のフローチャートである。

【図29】図24のパケットを受信した時に起動される処理のフローチャートである。

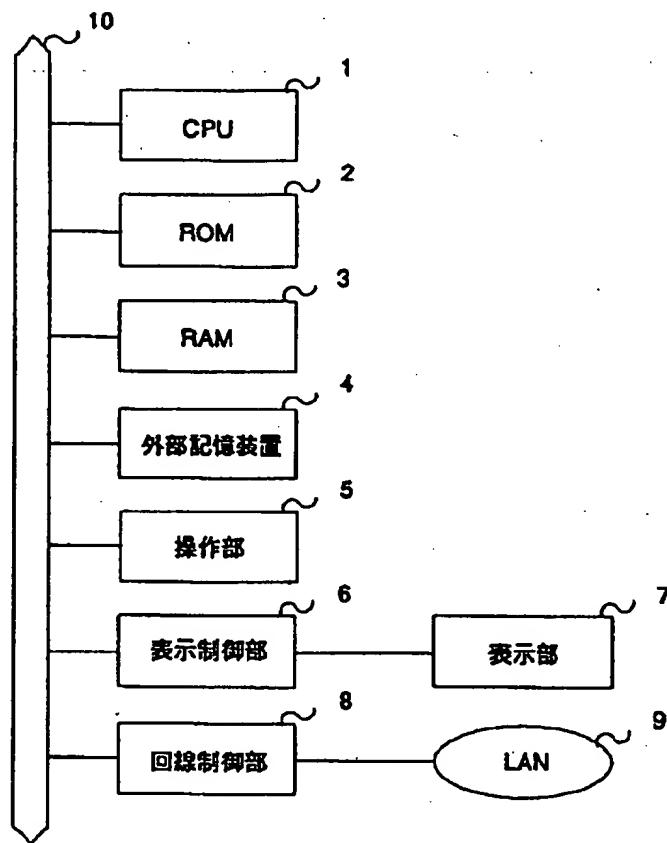
【符号の説明】

1—CPU

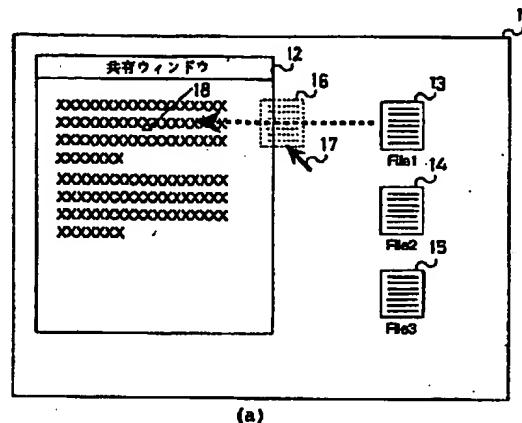
2 ROM

- 3 RAM
- 4 外部記憶装置
- 5 操作部
- 6 表示制御部
- 7 表示部
- 8 回線制御部
- 9 回線
- 10 バス

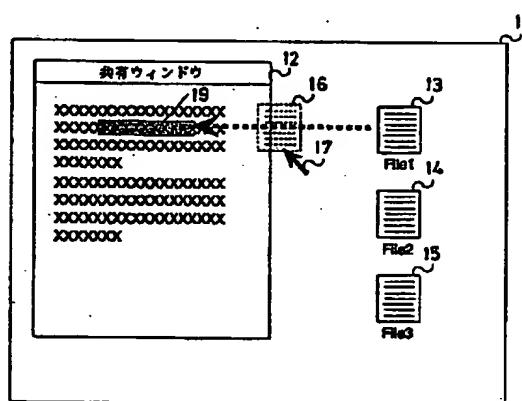
【図1】



【図2】

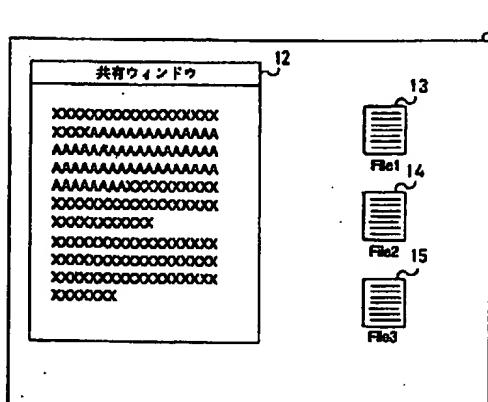


(a)

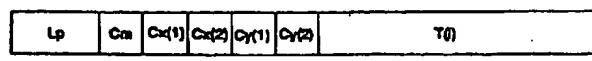


(b)

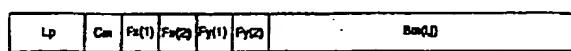
【図3】



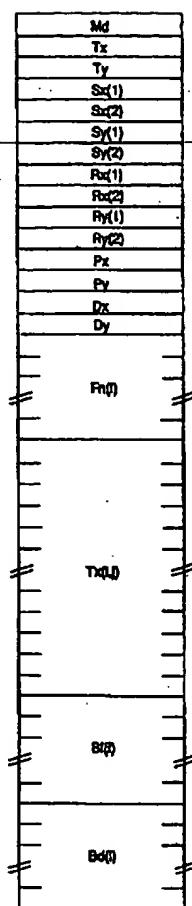
【図5】



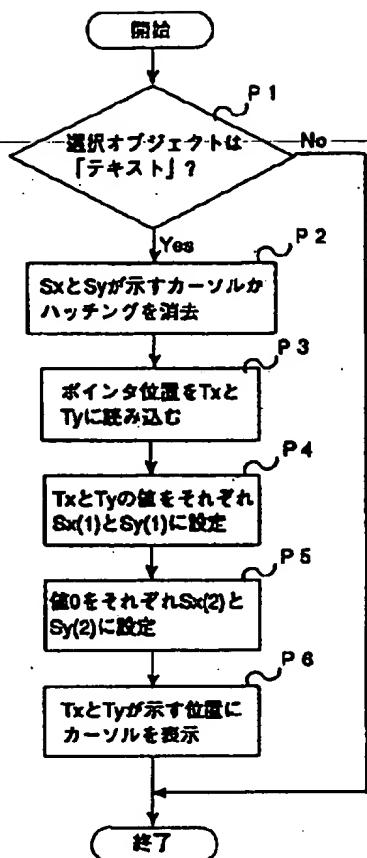
【図14】



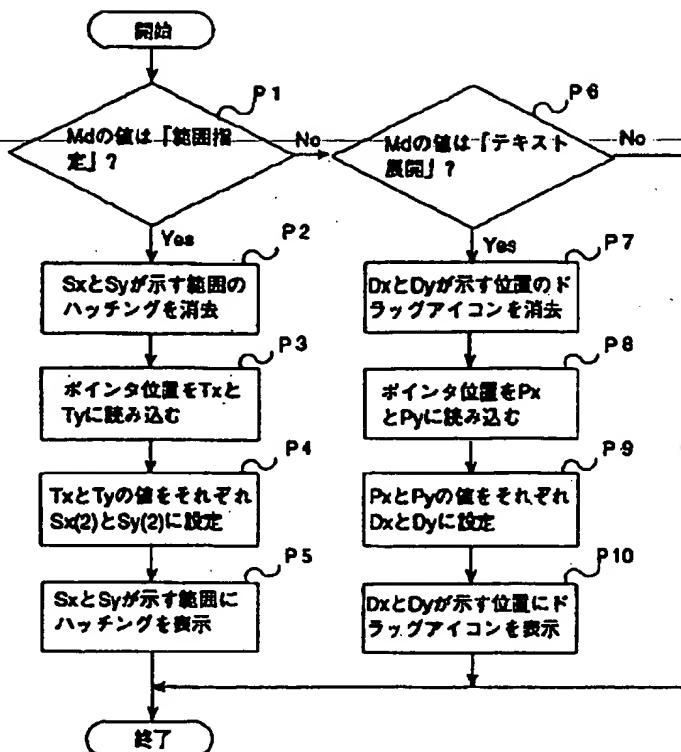
【図4】



【図6】



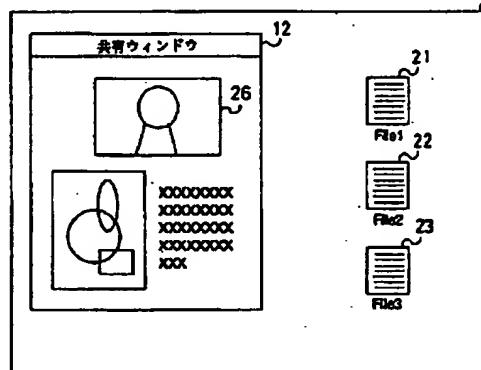
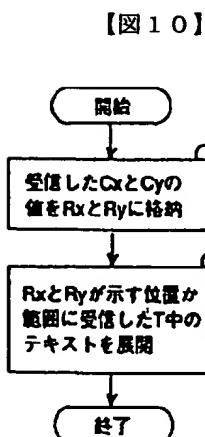
【図8】



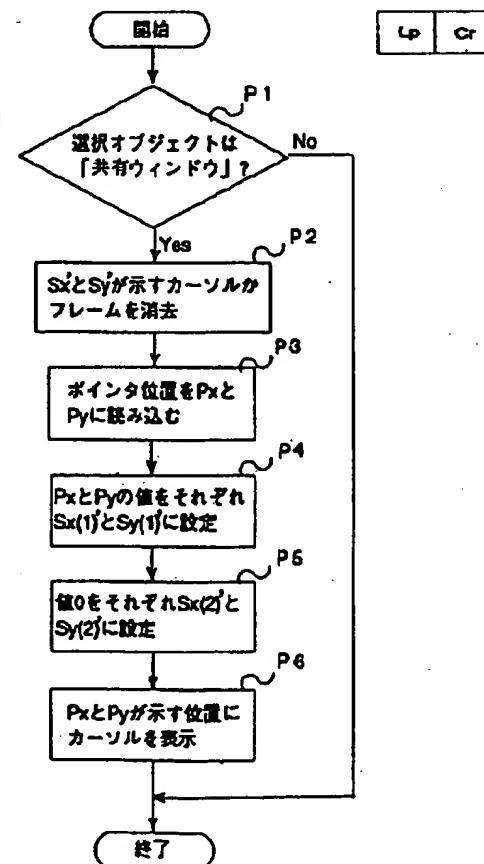
【図15】

【図24】

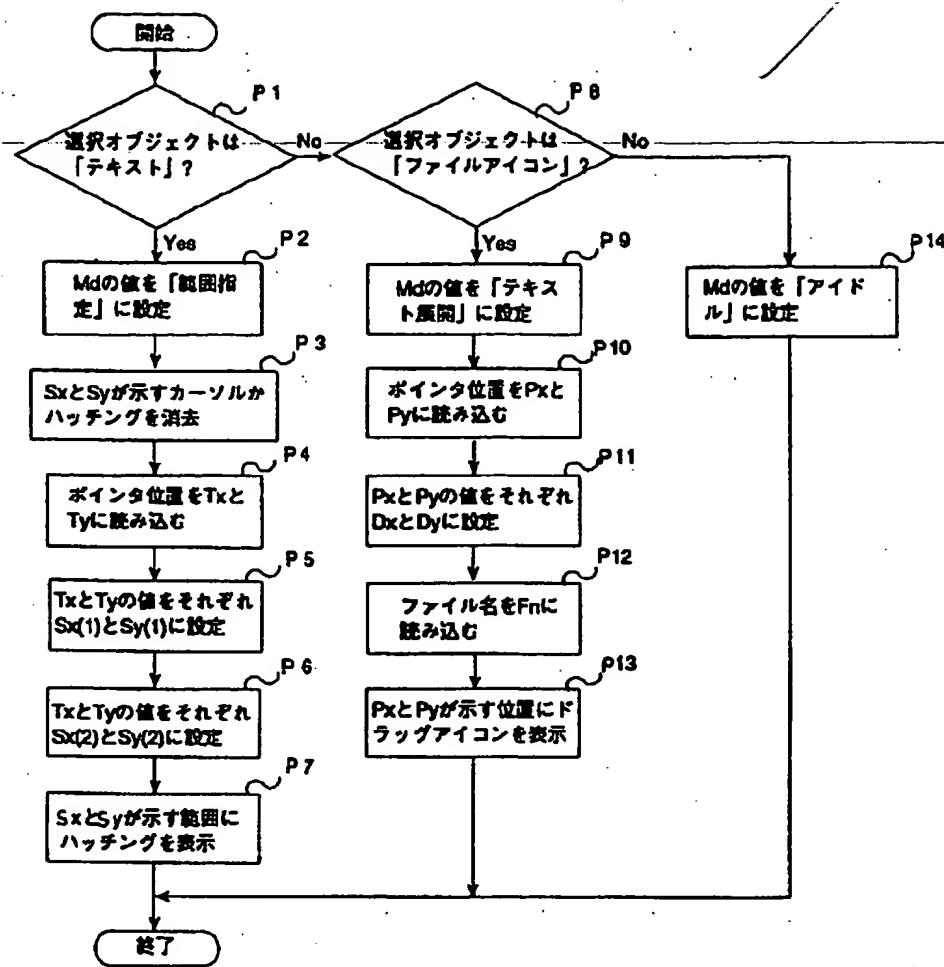
【図12】



【図23】



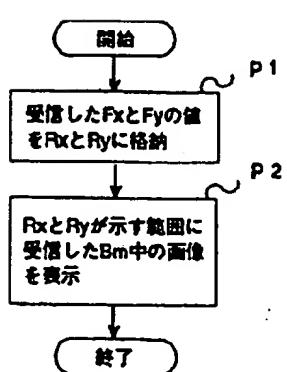
【図 7】



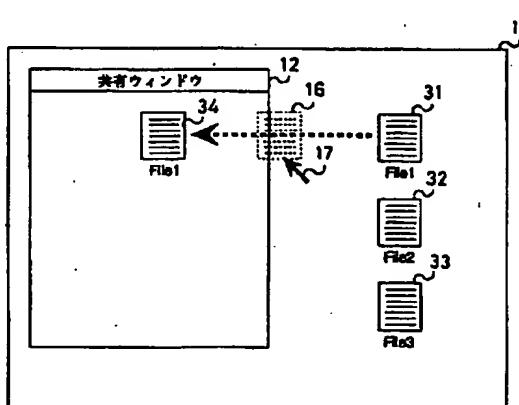
【図 13】

Md'
Px
Py
Wx(1)
Wx(2)
Wy(1)
Wy(2)
Sx(1)
Sx(2)
Sy(1)
Sy(2)
Rx(1)
Rx(2)
Ry(1)
Ry(2)
Dx
Dy
Fn()
Bx
By
Bx(L)
Bx
By
Bx(L)
Bdx
Bdy
Bdx(L)

【図 19】



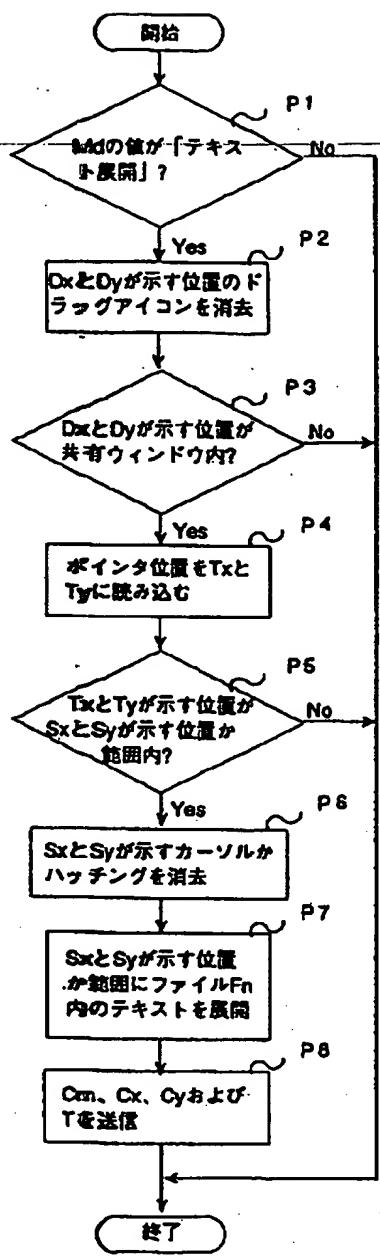
【図 20】



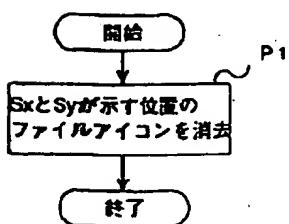
【図 22】

Md'
Px
Py
Dx
Dy
Cx
Cy
Sx
Sy
Rx
Ry
Fx
Fy
Bx
Bd

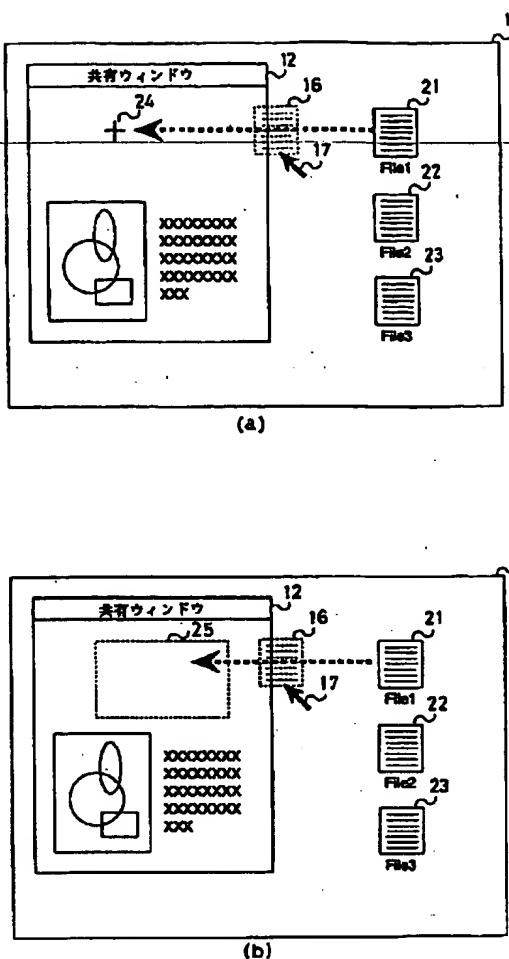
【図9】



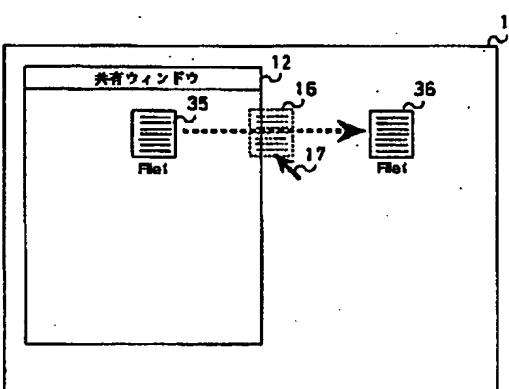
【図29】



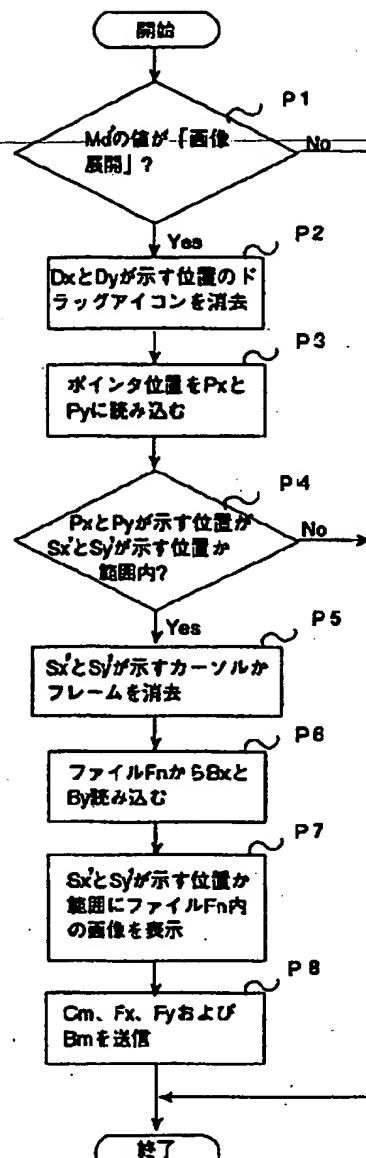
【図11】



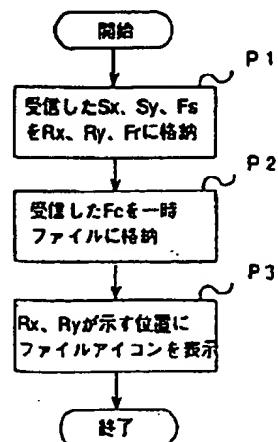
【図21】



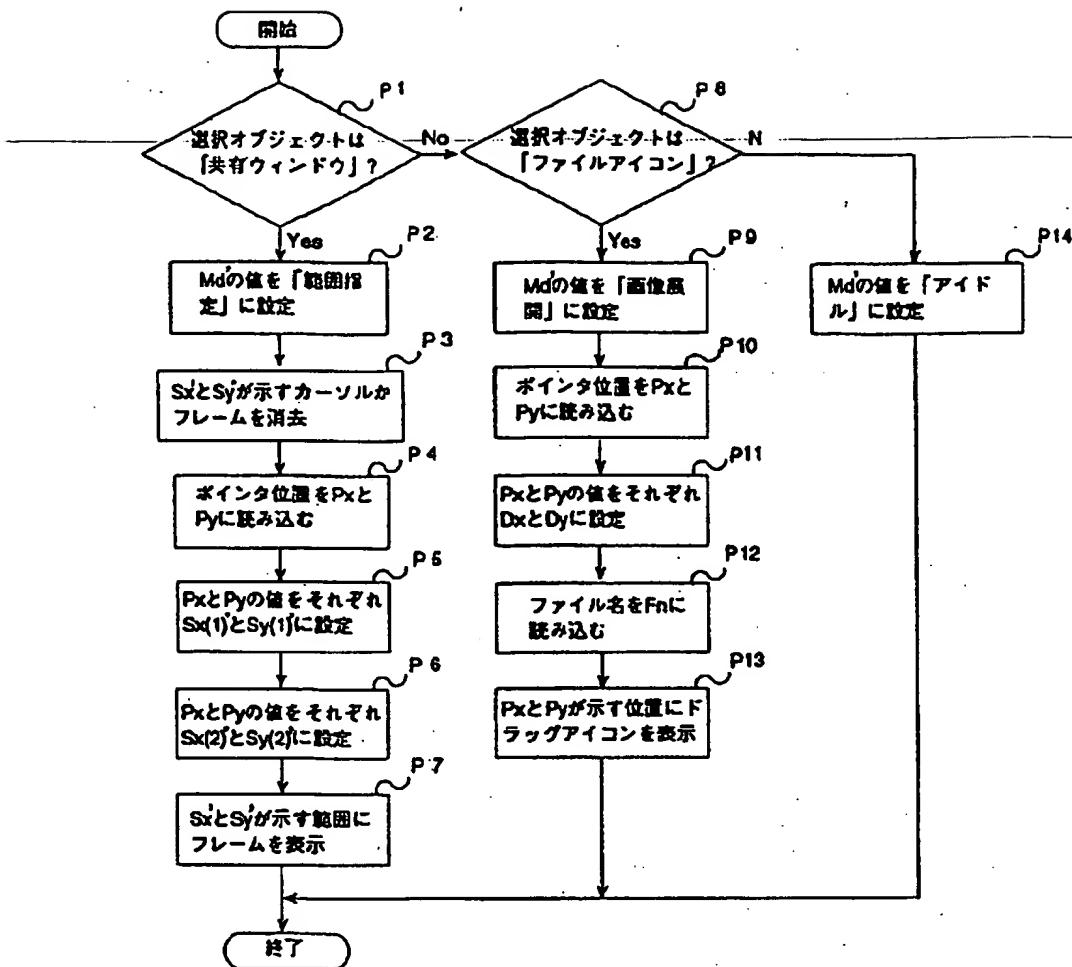
【図18】



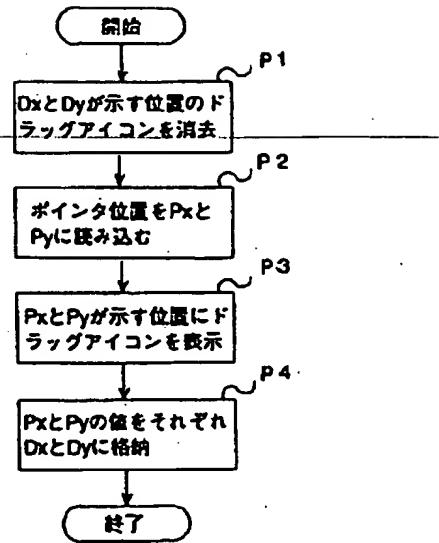
【図28】



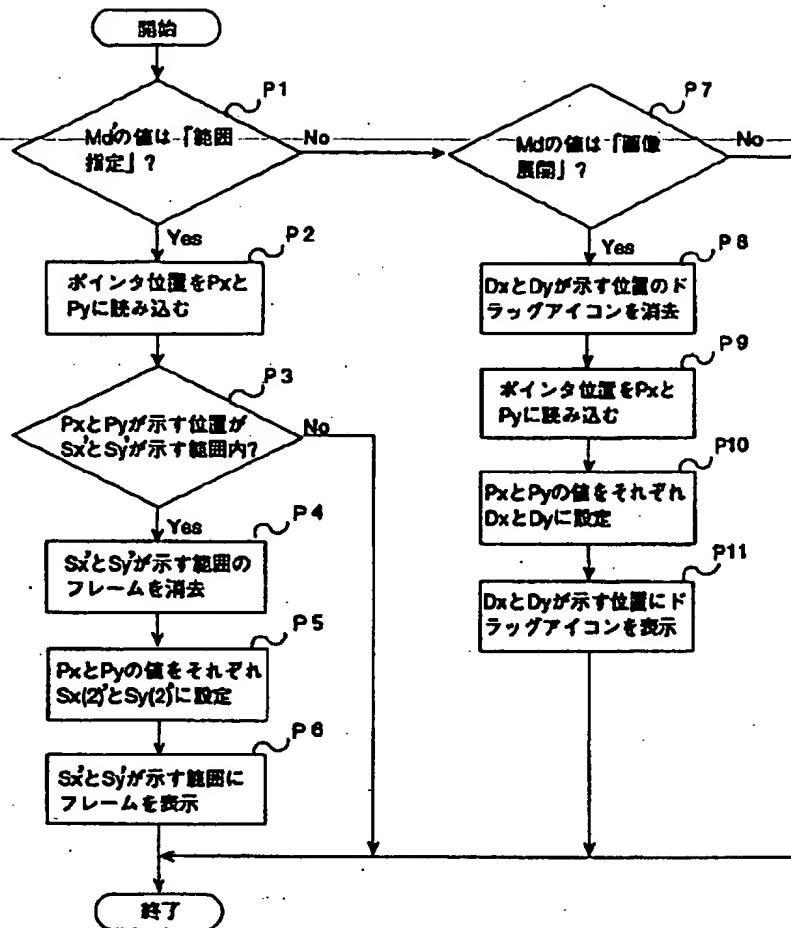
【図16】



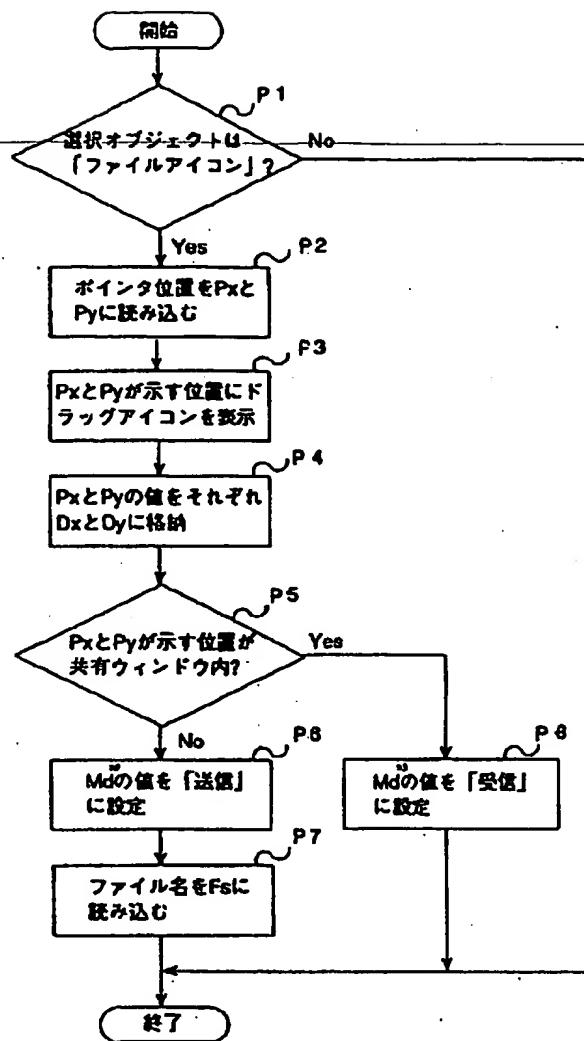
【図26】



【図17】



【図25】



【図27】

